aws

Guía para desarrolladores

Amazon Cognito



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Cognito: Guía para desarrolladores

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Amazon Cognito? 1
¿Qué partes de Amazon Cognito necesita? 1
Qué puede crear con Amazon Cognito 2
Grupos de usuarios 4
Grupos de identidades 5
Características de Amazon Cognito 6
Grupos de usuarios 6
Grupos de identidades 9
Comparación de grupos de usuarios y grupos de identidades de Amazon Cognito 11
Introducción a Amazon Cognito 16
Disponibilidad regional 16
Precios de Amazon Cognito 17
Términos y conceptos 17
General 18
Grupos de usuarios 21
Grupos de identidades 25
Introducción al AWS 26
Regístrese para obtener una Cuenta de AWS 27
Introducción 28
Tu primer inicio de sesión 29
Su primera aplicación y grupo de usuarios 30
Otras opciones de aplicación 33
Ejemplo de SPA de React 34
Ejemplo de aplicación móvil de Flutter 37
Siguientes pasos 40
Adición de un proveedor de redes sociales 41
Adición de un IdP SAML 50
Opciones de introducción adicionales 52
¿Planeando su diseño 55
Elija un enfoque 55
Situaciones comunes de Amazon Cognito 57
Autenticar con un grupo de usuarios 58
Acceso a los recursos del servidor 59
Acceso a los recursos con API Gateway y Lambda 60

Acceda a AWS los servicios con un grupo de usuarios y un grupo de identidades 60
Autenticate con un tercero y accede AWS servicios con un grupo de identidades 61
Acceso AWS AppSync recursos con Amazon Cognito 62
Machine-to-machine (M2M) 63
Step-up authentication 63
Configurar tu superficie de inicio de sesión 65
Inicio de sesión administrado 65
Localización de inicio de sesión administrado 67
Documentos de términos 69
Configuración del inicio de sesión administrado con la consola de Amazon Cognito. 70
Configurar el inicio de sesión gestionado con AWS Amplify 72
Consulta de la página de inicio de sesión 72
Personalización de las páginas de autenticación 74
Cosas que debe saber sobre el inicio de sesión administrado y la interfaz de usuario alojada 74
Configuración de un dominio 78
Creación de marca y personalización 94
Migrar a un inicio de sesión gestionado 109
Autenticación de usuarios 111
Características 111
Sign-up 112
Sign-in 113
Inicio de sesión administrado 114
Protección contra amenazas 116
Personalizar la experiencia del usuario 116
Monitoreo y análisis 116
Integración de los grupos de identidades de Amazon Cognito 116
Planes de características de grupo de usuarios 117
Selección de un plan de características 119
Características por plan 120
Características del plan Essentials 124
Plan de características Plus 128
Desactivación de las características no aptas 132
Autenticación 133
Clientes de aplicaciones 135
Implementación de flujos de autenticación 146

Cosas que debe saber ... 149
Ejemplo de flujo de autenticación ... 152
Autenticación del inicio de sesión administrado ... 155
Autenticación de SDK ... 159
Flujos de autenticación ... 166
Modelos de autorización de SDK ... 194
Third-party Inicio de sesión con IdP ... 210
Acceso a recursos de ... 289
Acceso a los recursos con Verified Permissions ... 290
Acceso a los recursos de API Gateway ... 292
Acceder a AWS los recursos mediante un grupo de identidades ... 293
Tokens de los grupos de usuarios ... 299
Tokens de ID ... 301
Cabecera del token de identificación ... 301
Carga útil predeterminada del token de ID ... 302
Identificador, token, firma ... 310
Tokens de acceso ... 310
Encabezado del token de acceso ... 311
Carga útil predeterminada del token de acceso ... 312
Machine-to-machine carga útil del token de acceso ... 315
Firma del token de acceso ... 315
Analice los tokens de forma segura ... 316
Tokens de actualización ... 317
Actualizar tokens ... 317
Rotación de tokens de actualización ... 318
Actualización de los tokens de API y SDK ... 320
Actualización del token de OAuth ... 322
Revocación de los tokens de actualización ... 322
Revocación de tokens ... 324
Habilitar la revocación de tokens ... 325
Revocación de un token ... 326
Verificación de un token web JSON ... 326
Requisitos previos ... 327
Validación de los tokens con aws-jwt-verify ... 328
Descripción e inspección de tokens ... 329
Almacenamiento en caché de tokens ... 335

Almacenamiento en caché de tokens de acceso de máquina a máquina con Amazon API Gateway ... 336
M2M y ámbitos ... 339
Autorización de API ... 340
Machine-to-machine autorización (M2M) ... 340
Acerca de los ámbitos ... 343
Acerca de los servidores de recursos ... 345
Vinculación de recursos ... 350
Administración de usuarios ... 352
Administración de usuarios ... 352
Permitir el registro de usuarios ... 353
Inscripción y confirmación de cuentas de usuario ... 356
Creación de usuarios como administrador ... 387
Agregar grupos a un grupo de usuarios ... 395
Gestión y búsqueda de usuarios ... 398
Contraseñas ... 406
Importación de usuarios a un grupo de usuarios ... 413
Atributos ... 441
Actualización del grupo de usuarios y del cliente de aplicación ... 461
Ajustes que no se pueden cambiar ... 462
Configuración de SMS ... 463
Actualización de un grupo de usuarios con un AWS SDK, AWS CDK, o API REST ... 464
Personalización de grupos de usuarios ... 467
Disparadores Lambda ... 467
Cosas que debe saber ... 470
Configuración de desencadenadores ... 473
Evento desencadenador de Lambda para un grupo de usuarios ... 473
Parámetros comunes del desencadenador de Lambda para un grupo de usuarios ... 474
Metadatos del cliente ... 476
Activación de los orígenes por operación ... 479
Orígenes del desencadenador por función ... 486
Antes del registro ... 492
Después de la confirmación ... 499
Antes de la autenticación ... 502
Después de la autenticación ... 506
Federación entrante ... 510

Desafío personalizado ... 521
Antes de la generación del token ... 544
Migración de usuario ... 568
Mensaje personalizado ... 575
Remitentes personalizados ... 584
Mensajería y análisis ... 601
Configuración del correo electrónico ... 601
Configuración de mensajes SMS ... 617
Uso del análisis de Amazon Pinpoint ... 634
Proteger los grupos de usuarios ... 641
Prácticas recomendadas de seguridad ... 643
Protección de un grupo de usuarios en el nivel de red ... 643
Protección contra el abuso de los mensajes SMS ... 643
Funcionamiento de la autenticación pública ... 643
Protección de clientes confidenciales con secretos de cliente ... 647
Protección de otros secretos ... 648
Privilegio mínimo de administración del grupo de usuarios ... 649
Protección y verificación de tokens ... 652
Determinación de los proveedores de identidad confiables ... 652
El efecto de los ámbitos en el acceso a los perfiles de usuario ... 652
Desinfección de las entradas para los atributos de usuario ... 653
Adición de MFA ... 653
Cosas que debe saber ... 655
Preferencias de MFA del usuario ... 659
Lógica de MFA ... 662
Configurarción de MFA ... 667
MFA con SMS y correo electrónico ... 668
MFA con token de software TOTP ... 670
Niveles de autenticación con notificaciones de ACR y AMR ... 678
Niveles y métodos de autenticación ... 679
Configuración de los nombres de los niveles de ACR ... 682
Requisitos del plan de funciones ... 684
Sus responsabilidades a la hora de aumentar la autenticación ... 684
Usar dispositivos ... 685
Configuración de dispositivos recordados ... 685
Obtención de la clave del dispositivo ... 687

Inicio de sesión con un dispositivo ... 688
Visualización, actualización y olvido de dispositivos ... 690
Protección contra amenazas ... 691
Consideraciones y limitaciones en la protección contra amenazas ... 694
Activación de la protección contra amenazas en los grupos de usuarios ... 695
Conceptos de aplicación de la protección contra amenazas ... 696
Opciones de protección contra amenazas ... 697
Requisitos previos ... 698
Configuración de la protección contra amenazas ... 698
Credenciales comprometidas ... 705
Autenticación flexible ... 706
Recopilación de datos ... 718
AWS WAF ACL web ... 722
Cosas que debe saber AWS WAF ACL web y Amazon Cognito ... 723
Content-Security-Policy en las respuestas de bloque personalizadas ... 725
Los códigos de estado de las respuestas de bloqueo personalizadas ... 726
Asociación de una ACL web con un grupo de usuarios ... 727
Probando y registrando AWS WAF ACL web ... 729
Sensible a mayúsculas y minúsculas ... 730
Protección contra eliminación ... 732
Administración de divulgación de usuarios ... 734
Operaciones de autenticación y creación de usuarios ... 735
Operaciones de restablecimiento de contraseña ... 740
Operaciones de confirmación ... 741
Multi-tenancy mejores prácticas ... 742
Per-tenant grupos de usuarios ... 744
Per-tenant clientes de aplicaciones ... 746
Per-tenant grupos de usuarios ... 748
Per-tenant atributos personalizados ... 750
Per-tenant ámbitos personalizados ... 752
Ejemplo de recurso ... 755
Multi-tenancy recomendaciones de seguridad ... 756
Integración con aplicaciones ... 758
Autenticación con AWS Amplify ... 760
Creación de una interfaz de usuario (IU) con Amplify ... 760
Autenticación con AWS SDK ... 761

Funcionamiento de la autenticación ... 762
Autenticación del inicio de sesión administrado ... 763
Autenticación de SDK ... 766
Third-party autenticación del proveedor de identidad ... 769
Autenticación del grupo de identidades ... 772
Trabajando con AWS SDK ... 775
Autorización con Amazon Verified Permissions ... 776
Autorización de API con Verified Permissions ... 778
Política de ejemplo para un usuario de Amazon Cognito ... 781
Referencia de puntos de enlace del grupo de usuarios ... 784
Puntos de conexión en el inicio de sesión administrado ... 785
Punto de conexión Login ... 789
Punto de conexión Logout ... 791
Puntos de conexión de la federación ... 795
Emisor de OIDC ... 797
Autorizar punto de conexión ... 800
Punto de conexión de token ... 813
El punto de conexión userInfo ... 826
Revocación de puntos de conexión ... 830
El punto de conexión idpresponse de SAML ... 832
Concesiones de OAuth 2.0 ... 834
Uso de la PKCE ... 836
Cómo utiliza Amazon Cognito la PKCE ... 836
Respuestas de error de federación y de inicio de sesión administrado ... 838
Despliegue y operación ... 841
Cuotas ... 841
Descripción de las cuotas de la tasa de solicitudes de la API ... 842
Administración de las cuotas de la tasa de solicitudes de la API ... 845
Cuotas de tasa de solicitudes de grupos de usuarios ... 854
Cuotas de tasa de solicitudes de grupos de identidades ... 869
Cuotas sobre el número y el tamaño de los recursos ... 871
Multi-Region replicación ... 879
Lo que debe saber sobre la replicación multirregional ... 880
Limitaciones de la replicación multirregional ... 881
Configuración de la replicación multirregional ... 881
Operaciones de API compatibles en regiones secundarias ... 884

La conmutación por error en grupos de usuarios de varias regiones 886
Etiquetado de recursos 889
Recursos admitidos 890
Restricciones de las etiquetas 890
Administración de etiquetas con la consola 891
AWS CLI ejemplos 891
Acciones de API de 895
CloudFormation etiquetado 895
Resolución de problemas 896
Errores en la configuración de dominios personalizados 896
Error de `token de actualización no válido` 899
Errores de respuesta SAML no válidos en la federación 900
Los usuarios de inicio de sesión administrado no pueden seleccionar un factor MFA 903
No se puede recibir el código de restablecimiento de contraseña a través de email/SMS 904
El restablecimiento de la contraseña no se puede restablecer debido a los atributos de recuperación no verificados: `no se ha podido restablecer la contraseña de la cuenta, póngase en contacto con el servicio de asistencia` o inténtelo de nuevo 905
`Errores de SECRET_HASH` 906
La consola de Amazon Cognito elige una configuración predeterminada para un nuevo grupo de usuarios 906
Recursos adicionales de solución de problemas 907
Ejemplos de código 908
Amazon Cognito Identity 910
Conceptos básicos 911
Escenarios 936
Amazon Cognito Identity Provider 938
Conceptos básicos 940
Escenarios 1112
Amazon Cognito Sync 1300
Conceptos básicos 1300
Intercambio de identidades por credenciales AWS 1303
Grupos de identidades de Amazon Cognito 1303
Configuración de grupos de identidades 1305
Flujo de autenticación de grupos de identidades 1313
Roles de IAM 1324

Prácticas recomendadas de seguridad 1344
Uso de atributos para el control de acceso 1348
Uso del control de acceso basado en roles 1357
Obtención de credenciales 1364
Uso de credenciales 1371
Third-party proveedores de identidad 1373
Developer-authenticated identidades 1407
Cambio de identidades 1425
Introducción a los grupos de identidades 1428
Creación de un grupo de identidades en Amazon Cognito 1429
Configurar un SDK 1431
Integración de los proveedores de identidad 1431
Obtención de credenciales 1431
Aplicación de ejemplo 1432
Seguridad 1467
Protección de datos 1468
Cifrado de datos 1469
Cifrado en tránsito 1478
Identity and Access Management 1479
Público 1480
Autenticación con identidades 1480
Administración del acceso con políticas 1482
Cómo funciona Amazon Cognito con IAM 1483
Identity-based ejemplos de políticas 1492
Resolución de problemas 1497
Cómo utilizar roles vinculados a servicios 1499
Registro y supervisión 1504
Supervisión de costos 1505
Exportación de registros de grupos de usuarios 1508
Supervisión de cuotas y uso 1519
CloudTrail registros 1541
AWS PrivateLink 1569
Flujos de autenticación para AWS PrivateLink integración 1570
Modos operativos para AWS PrivateLink 1572
Consideraciones 1573
Controlar el acceso con políticas de control de recursos 1578

Creación de un punto de conexión de interfaz 1579
Creación de una política de punto de conexión 1580
Cree una política basada en la identidad 1582
Validación de conformidad 1583
Resiliencia 1584
Consideraciones de datos regionales 1585
Seguridad de la infraestructura 1586
Configuración y análisis de vulnerabilidades 1586
AWS políticas administradas 1586
Actualizaciones de políticas 1588
Legacy 1592
Amazon Cognito Sync 1592
Cambio de disponibilidad de Amazon Cognito Sync 1593
Introducción a Amazon Cognito Sync 1594
Sincronización de datos entre clientes 1596
Gestión de las devoluciones de llamada de eventos 1606
Implementación de la sincronización mediante inserción 1624
Implementación de flujos de Amazon Cognito Sync 1634
Personalización de los flujos de trabajo con Amazon Cognito Events 1637
Uso de Amazon Cognito Sync con grupos de identidades 1642
Administración de conjuntos de datos 1642
Publicación en masa de datos 1644
Activar sincronización mediante inserción 1644
Configuración de Amazon Cognito Streams 1645
Configuración de Amazon Cognito Events 1645
Marca de interfaz de usuario alojada (clásica) 1645
Especificación de un logotipo personalizado en la marca clásica 1646
Especificación de personalizaciones con CSS en la creación clásica de marca 1646
Personalización de la interfaz de usuario alojada con la marca clásica en Consola de administración de AWS 1651
Personalizar la interfaz de usuario alojada con la marca clásica en la API de grupos de usuarios y con AWS CLI 1652
Historial de revisión 1653
..... mdclxxxii

¿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

- ¿Qué partes de Amazon Cognito necesita?
- Qué puede crear con Amazon Cognito
- Grupos de usuarios
- Grupos de identidades
- Características de Amazon Cognito
- Comparación de grupos de usuarios y grupos de identidades de Amazon Cognito
- Introducción a Amazon Cognito
- Disponibilidad regional
- Precios de Amazon Cognito
- Términos y conceptos comunes de Amazon Cognito
- Introducción al AWS

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

¿Qué partes de Amazon Cognito necesita?

Amazon Cognito responde a dos preguntas diferentes. Elija en función de lo que debe hacer su aplicación.

¿Quién es este usuario? (autenticación)

> Utilice un grupo de usuarios. Un grupo de usuarios es un directorio de usuarios que se basa en el estándar OpenID Connect (OIDC) para autenticar a los usuarios y emitir tokens web JSON (JWT) a tu aplicación o API. Los usuarios pueden iniciar sesión directamente, a través de proveedores

sociales como Google y Apple, o a través de proveedores empresariales de SAML y OIDC. La mayoría de las aplicaciones que necesitan registrarse e iniciar sesión comienzan aquí.

¿A qué AWS recursos puede acceder este usuario? (autorización)

Utilice un grupo de identidades. Un grupo de identidades intercambia un token de confianza (de un grupo de usuarios, un proveedor social o un proveedor de SAML u OIDC) por AWS credenciales temporales con privilegios limitados para que la aplicación pueda llamar a servicios como Amazon S3 y Amazon DynamoDB en nombre del usuario. Un grupo de identidades también puede emitir credenciales con privilegios limitados a los usuarios invitados que no hayan iniciado sesión sin ningún token.

¿Necesitas ambas?

Úsalos juntos cuando tu aplicación inicie sesión con los usuarios y necesite concederles acceso directo a los AWS recursos: autentícate con un grupo de usuarios y, a continuación, intercambia el token del grupo de usuarios por un grupo de identidades a cambio de AWS credenciales.

Ninguno de los componentes requiere el otro. Un grupo de usuarios puede emitir tokens para tu propio back-end sin un grupo de identidades. Un grupo de identidades puede negociar las credenciales de un proveedor que ya tiene, incluso sin un grupo de usuarios. Para obtener más información sobre las diferencias entre los dos componentes, consulteComparación de grupos de usuarios y grupos de identidades de Amazon Cognito. Para empezar a crear, consulteIntroducción a Amazon Cognito.

Qué puede crear con Amazon Cognito

La mayoría de las aplicaciones de Amazon Cognito tienen algunas formas comunes. Utilice esta lista para encontrar la que se parezca más a lo que está creando y, a continuación, siga el enlace para ver el tutorial completo.

Inicia sesión con los usuarios en tu aplicación

Un grupo de usuarios autentica a tus usuarios y emite tokens web JSON (JWT) que tu aplicación o API verifica. Este es el punto de partida para la mayoría de las aplicaciones que necesitan registrarse e iniciar sesión. Consulte Autenticar con un grupo de usuarios.

Proteja su propia API o sus recursos de back-end

Utilice el token de acceso de un grupo de usuarios para autorizar las solicitudes a los recursos del lado del servidor o deje que Amazon API Gateway valide el token y controle el acceso a una

función de Lambda o a su propia API. Consulte Acceso a los recursos de backend con tokens de grupos de usuarios y Acceso a los recursos con API Gateway y Lambda mediante un grupo de usuarios.

Ofrezca a los usuarios acceso temporal a los servicios AWS

Cambie un token de grupo de usuarios por un grupo de identidades por AWS credenciales temporales con privilegios limitados para que su aplicación pueda llamar a servicios como Amazon S3 y Amazon DynamoDB en nombre del usuario. Consulte Acceso AWS servicios con un grupo de usuarios y un grupo de identidades.

Permite a los usuarios iniciar sesión con Google, Apple o un directorio empresarial

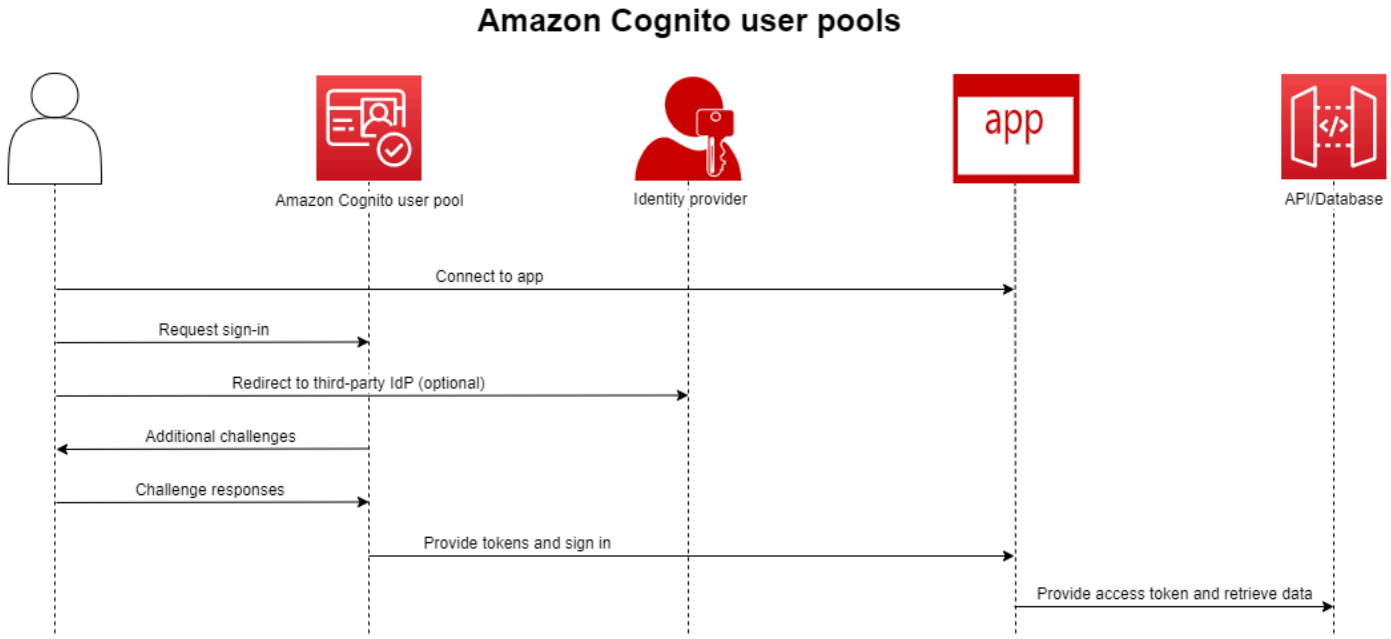
Federate a través de una red social o un SAML/OIDC proveedor. Un grupo de usuarios gestiona el inicio de sesión y emite a tu aplicación un conjunto estándar de tokens, o un grupo de identidades intercambia el token externo directamente por AWS credenciales. Consulte Autenticate con un tercero y accede AWS servicios con un grupo de identidades.

Autoriza una aplicación o un dispositivo sin un usuario humano

Machine-to-machine La autorización (M2M) emite un token de acceso a una aplicación, servicio o dispositivo que se autentica como tal con la `client_credentials` concesión, sin que ningún usuario humano esté al tanto. Consulte Autoriza aplicaciones de máquina a máquina (M2M).

Para ver el conjunto completo de escenarios con diagramas de arquitectura, consulte. Situaciones comunes de Amazon Cognito

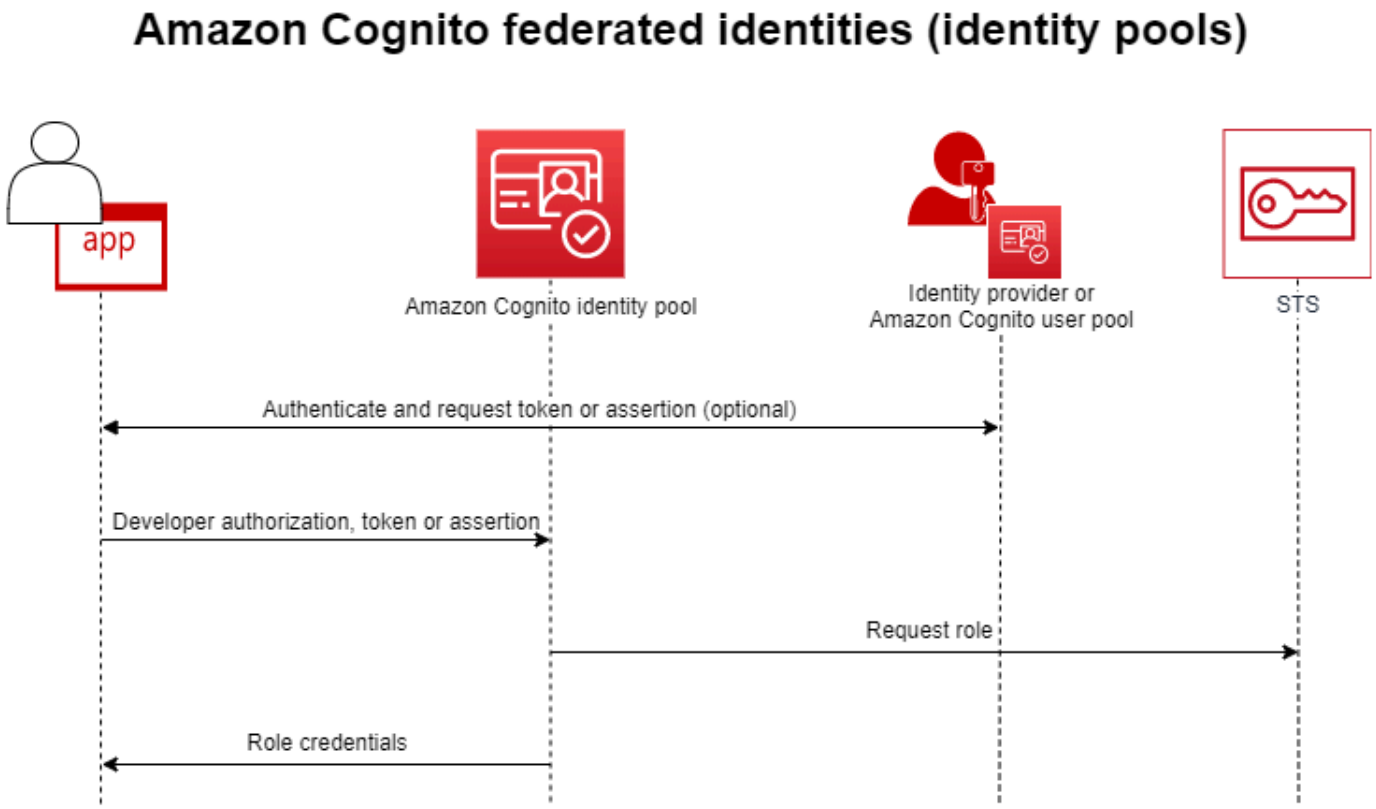
Grupos de usuarios



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. Puede proporcionar un inicio de sesión único (SSO) en su aplicación para las identidades de los empleados de su organización en SAML 2.0 y OIDC con grupos de usuarios. IdPs 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. Situaciones comunes de Amazon Cognito

En Amazon Cognito, la obligación de seguridad de la nube del modelo de responsabilidad compartida cumple con los requisitos SOC 1-3, PCI DSS e ISO 27001. HIPAA-BAA Puede diseñar su seguridad en la nube en Amazon Cognito para que cumpla con los estándares SOC 1-3, ISO 27001 y PCI DSS HIPAA-BAA, pero no con PCI DSS. Para obtener más información, consulte Servicios de AWS en el ámbito. Véase 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 Consola de administración de AWS, 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 de usuario en un JWT que ellos envían a su 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	Description (Descripción)
Proveedor de identidades 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
Proveedor de servicios SAML 2.0	Transforma las afirmaciones de SAML en identificadores y identificadores de acceso

Parte que confía en el OIDC	Transforma los tokens OIDC en tokens de identificación y acceso
Proveedor social: parte confiante	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 un inicio de sesión gestionado
Soporte de API para su propia interfaz de usuario	Cree, gestione y autentique usuarios mediante solicitudes de API de autenticación en los SDK compatibles[1] AWS
Multi-factor authentication	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 inseguros[1]
Personalice los flujos de autenticación	Cree su propio mecanismo de autenticación o añada pasos personalizados a los flujos existentes[2]
Groups	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	Personaliza tu ID y accede a los tokens con reclamos nuevos, modificados y suprimidos
Personalice los atributos del usuario	Asigna valores a los atributos de usuario y añade tus propios atributos personalizados

[1] La característica no está disponible para los usuarios federados.

[2] Esta característica no está disponible para usuarios federados ni con inicio de sesión administrado.

Para obtener más información sobre los grupos de usuarios, consulte Creación de una nueva aplicación en la consola de Amazon Cognito 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 una prueba de autenticación personalizada con Developer-authenticated identidades. Puede conceder credenciales de AWS temporales a usuarios invitados con identidades no autenticadas.

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

Role-based control de acceso

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	Description (Descripción)
Parte que confía en el grupo de usuarios de Amazon Cognito	Cambie un token de identificación de su grupo de usuarios por credenciales de identidad web de AWS STS
Proveedor de servicios SAML 2.0	Intercambie las afirmaciones de SAML para obtener credenciales de identidad web desde AWS STS
Parte de confianza del OIDC	Cambie los tokens OIDC por credenciales de identidad web desde AWS STS
Proveedor social: parte de confianza	Intercambia los tokens de OAuth de Amazon, Facebook, Google, Apple y Twitter por credenciales de identidad web de AWS STS
Parte de confianza personalizada	Con AWS las credenciales, puede intercambiar solicitudes en cualquier formato por credencia les de identidad web desde AWS STS
Acceso sin autenticar	Emita credenciales de identidad web de acceso limitado sin autenticación AWS STS
Role-based control de acceso	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 grupo de identidad es
Attribute-based control de acceso	Convierta las notificaciones en etiquetas principales para su sesión AWS STS temporal y utilice 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	Description (Descripción)	Grupos de usuarios	Grupos de identidades
Proveedor de identidades OIDC	Emita tokens de ID de OIDC para autenticar a los usuarios de la aplicación	✓	
Directorio de usuarios	Almacene los perfiles de usuario para la autenticación	✓	
Autoriza el acceso a la API	Emita tokens de acceso para autorizar el acceso de los usuarios a las API (incluidas las operaciones de API de autoservicio de los perfiles de usuario), las bases de datos y otros recursos que aceptan los ámbitos de OAuth	✓	

Autorización de identidad web de IAM	Genera fichas con las que puedas canjearlas AWS STS por credenciales temporales AWS		✓
Proveedor de servicios SAML 2.0 y proveedor de identidad OIDC	Emita tokens OIDC personalizados en función de las afirmaciones de un proveedor de identidad de SAML 2.0	✓	
Parte de confianza del OIDC y proveedor de identidad del OIDC	Emita tokens OIDC personalizados en función de las afirmaciones de un proveedor de identidad del OIDC	✓	
Parte que confía en OAuth 2.0 y proveedor de identidad del OIDC	Emite tokens OIDC personalizados basados en los alcances de los proveedores sociales de OAuth 2.0, como Apple y Google	✓	
Proveedor de servicios y agente de credenciales SAML 2.0	Emita AWS credencia les temporales en función de las afirmaciones de un proveedor de identidad de SAML 2.0		✓

Persona de confianza y agente de credencia les del OIDC	Emita AWS credencia les temporales en función de las afirmaciones de un proveedor de identidad del OIDC		✓
Proveedor social, persona de confianza y agente de credencia les	Emita AWS credencia les temporales basadas en tokens web JSON de aplicaciones de desarrolladores con proveedores sociales como Apple y Google		✓
Agente de credencia les y agente de confianza del grupo de usuarios de Amazon Cognito	Emita AWS credencia les temporales basadas en tokens web JSON de grupos de usuarios de Amazon Cognito		✓
Agente de credencia les y persona de confianza personali zada	Emita AWS credencia les temporales para identidades arbitrari as, autorizadas por las credenciales de IAM del desarrollador		✓
Servicio frontend de autenticación	Registre, gestione y autentique a los usuarios con un inicio de sesión gestionado	✓	

Soporte de API para su propia interfaz de usuario de autenticación	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 inseguros[1]	✓
Personalice los flujos de autenticación	Cree su propio mecanismo de autenticación o añada pasos personalizados a los flujos existentes[1]	✓
Grupos de usuarios	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	Personaliza tu ID y accede a los tokens con pretensiones y alcances nuevos, modificados y suprimidos	✓

AWS WAF ACL web	Supervise y controle las solicitudes a su interfaz de autenticación con AWS WAF	✓	
Personalice los atributos del usuario	Asigna valores a los atributos de usuario y añade tus propios atributos personalizados	✓	
Acceso sin autenticar	Emita credenciales de identidad web de acceso limitado sin autenticación AWS STS		✓
Role-based control de acceso	Elija un rol de IAM para su usuario autenticado en función de sus afirmaciones y configure la confianza de su rol para limitar el acceso a los usuarios de identidad web		✓

Attribute-based control de acceso	Transforma las notificaciones 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 no está disponible para los usuarios federados.

Introducción a Amazon Cognito

Para ver ejemplos de aplicaciones de grupos de usuarios, consulte Creación de una nueva aplicación en la consola de Amazon Cognito.

Para ver una introducción a los grupos de identidades, consulte Introducción a los grupos de identidades de Amazon Cognito.

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

Para empezar a usar un AWS SDK, consulta las Herramientas para AWS desarrolladores. Si quiere ver los recursos para desarrolladores específicos de Amazon Cognito, consulte Recursos para desarrolladores de Amazon Cognito.

Para usar Amazon Cognito necesita una Cuenta de AWS. Para obtener más información, consulte Introducción al AWS.

Disponibilidad regional

Amazon Cognito está disponible en varias AWS regiones del 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 ver si Amazon Cognito está disponible actualmente en alguno de ellos Región de AWS, consulte AWS Servicios por región.

Para obtener más información sobre los puntos de enlace de los servicios de API regionales, consulte AWS Regiones y puntos de enlace en. 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.

Términos y conceptos comunes de Amazon Cognito

Amazon Cognito proporciona credenciales para aplicaciones móviles y web. Se basa en términos que son de uso común en la administración de identidades y accesos. Tiene a su disposición numerosas guías sobre los términos de identidad y acceso universales. Algunos ejemplos son:

- La terminología de la página Terminology in the IDPro Body of Knowledge
- AWS Servicios de identidad
- El glosario de CSRC de NIST

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 administración de identidades y accesos. No se trata de una lista exhaustiva de términos, sino simplemente una guía sobre su contexto de Amazon Cognito específico en esta guía.

Token de acceso

Un token web JSON (JWT) que contiene información sobre la autorización de una entidad para acceder a los sistemas de información.

Aplicación o app

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

Attribute-based control de acceso (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 los tokens de identificación en los grupos de usuarios y las etiquetas de entidades principales en los grupos de identidades.

Autenticación

El proceso para establecer una identidad auténtica con el fin de acceder a un sistema de información. Amazon Cognito acepta pruebas de autenticación de proveedores de identidad externos y también actúa como proveedor de autenticación para aplicaciones de software.

Autorización

El proceso para conceder permisos a un recurso. Los tokens de acceso de un grupo de usuarios contienen información que las aplicaciones pueden utilizar para permitir que los usuarios y los sistemas accedan a los recursos.

Servidor de autorización

Un sistema OAuth u OpenID Connect (OIDC) que genera tokens web JSON. El servidor de autorización administrado de grupos de usuarios de Amazon Cognito es el componente del servidor de autorización de los dos métodos de autenticación y autorización en los grupos de usuarios. Los grupos de usuarios también admiten los flujos de desafío-respuesta de la API en la autenticación de SDK.

Aplicación confidencial, aplicación de servidor

Aplicación a la que los usuarios se conectan de forma remota, con el código en un servidor de aplicaciones y acceso a secretos. Por lo general, se trata de una aplicación web.

Proveedor de identidades (IdP)

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)

JSON-formatted Documento que contiene afirmaciones sobre un usuario autenticado. Los tokens de ID 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

Machine-to-machine autorización (M2M)

El proceso para autorizar las solicitudes a los puntos de conexión de la API para entidades de máquina que no interactúan con el usuario, como un nivel de aplicación de servidor web. Los grupos de usuarios sirven la autorización M2M en concesiones de credenciales de cliente con ámbitos OAuth 2.0 en tokens de acceso.

Multi-factor autenticación (MFA)

Requisito por el que se exige a los usuarios que proporcionen una autenticación adicional después de proporcionar su nombre de usuario y contraseña. Los grupos de usuarios de Amazon Cognito cuentan con características de MFA para los usuarios locales.

Proveedor de OAuth 2.0 (redes sociales)

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)

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

clave de paso, WebAuthn

Forma de autenticación en la que las claves criptográficas, o claves de acceso, del dispositivo de un usuario proporcionan su prueba de autenticación. Los usuarios verifican que están

presentes con mecanismos biométricos o de código PIN en un autenticador de hardware o software. Las claves de acceso son resistentes a la suplantación de identidad y están vinculadas a determinadas características websites/apps, por lo que ofrecen una experiencia segura sin contraseñas. Los grupos de usuarios de Amazon Cognito admiten el inicio de sesión con claves de acceso.

Sin contraseña

Una forma de autenticación en la que el usuario no tiene que introducir una contraseña. Algunos de los métodos de inicio de sesión sin contraseña son las contraseñas de un solo uso (OTP), que se envían a direcciones de correo electrónico y números de teléfono, y las claves de acceso. Los grupos de usuarios de Amazon Cognito admiten el inicio de sesión con OTP y claves de acceso.

Aplicación pública

Aplicación autónoma en un dispositivo, con el código almacenado localmente y sin acceso a datos secretos. Por lo general, se trata de una aplicación móvil.

Servidor de recursos

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.

Role-based control de acceso (RBAC)

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

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

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

Proveedor SAML

IdP para un grupo de usuarios o un grupo de identidades que genera documentos de aserción firmados digitalmente y 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, pero no se ajustan a un formato de UUID específico,

incluido el UUID de RFC. No debe validar estrictamente el formato de los identificadores de Amazon Cognito.

Directorio de usuarios

Conjunto de usuarios y sus atributos que proporciona 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

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

Autenticación flexible

Característica de seguridad avanzada que detecta posibles actividades maliciosas y aplica medidas de seguridad adicionales a los perfiles de usuario.

Cliente de aplicación

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

URL de devolución de llamada, URI de redireccionamiento, URL de devolución

Una configuración en un cliente de aplicación y un parámetro en las solicitudes al servidor de autorización del grupo de usuarios. La URL de devolución de llamada es el destino inicial de los usuarios autenticados de su aplicación.

Choice-based authentication

Una forma de autenticación mediante API con grupos de usuarios en la que cada usuario tiene a su disposición un conjunto de opciones de inicio de sesión. Sus opciones pueden incluir el nombre de usuario y la contraseña con o sin MFA, el inicio de sesión con clave de acceso o el inicio de sesión con contraseñas de un solo uso por correo o SMS sin contraseña establecida. La aplicación puede configurar el proceso de elección de los usuarios solicitando una lista de opciones de autenticación o declarando una opción preferida.

Puede comparar esto con la autenticación basada en el cliente.

Client-based authentication

Una forma de autenticación con la API de los grupos de usuarios y los backends de las aplicaciones creados con AWS los SDK. En la autenticación declarativa, la aplicación determina

de forma independiente el tipo de inicio de sesión que debe realizar un usuario y lo solicita por adelantado.

Puede comparar esto con la autenticación basada en opciones.

Credenciales comprometidas

Característica 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 usuario.

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

Extensión de los procesos de autenticación con desencadenadores de Lambda que definen desafíos y respuestas adicionales para los usuarios.

Autenticación del dispositivo

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

Dominio, dominio del grupo de usuarios

Un dominio web que aloja sus páginas de inicio de sesión administrado en AWS. Puedes configurar el DNS en un dominio de tu propiedad o usar un prefijo de subdominio de identificación en un dominio que sea propietario. AWS

Plan Essentials

El plan de características con las últimas novedades en grupos de usuarios. El plan Essentials no incluye las características de seguridad de aprendizaje automático del plan Plus.

Proveedor externo

IdP que tiene una relación de confianza con un grupo de usuarios. Los grupos de usuarios actúan como una entidad intermedia entre los proveedores externos y su aplicación, y gestionan los procesos de autenticación con SAML 2.0, OIDC y los proveedores sociales. Los grupos de usuarios reúnen los resultados de autenticación de proveedores externos en un único IdP para que sus aplicaciones puedan procesar a muchos usuarios con una única biblioteca de OIDC que depende de ellos.

Plan de características

El grupo de características que puede seleccionar para un grupo de usuarios. Los planes de funciones tienen diferentes costos en tu factura. AWS Los grupos de usuarios nuevos están incluidos de forma predeterminada en el plan Essentials.

Planes actuales

- Plan Lite
- Plan Essentials
- Plan Plus

Usuario federado, usuario externo

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

Interfaz de usuario alojada (clásica), páginas de interfaz de usuario alojadas

La primera versión de los servicios de interfaz de autenticación, actor de confianza y proveedor de identidad en el dominio del grupo de usuarios. La interfaz de usuario alojada tiene un conjunto básico de características y una apariencia simplificada. Puede aplicar la marca de interfaz de usuario alojada cargando un archivo de imagen con el logotipo y un archivo con un conjunto predeterminado de estilos CSS. Puede comparar esto con el inicio de sesión administrado.

Desencadenador de Lambda

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

Usuario local

Perfil de usuario en el directorio de usuarios del grupo de usuarios que no se ha creado mediante la autenticación con un proveedor externo.

Usuario vinculado

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

Plan Lite

El plan de características con las características que se lanzaron originalmente con los grupos de usuarios. El plan Lite no incluye las nuevas características del plan Essentials ni las características de seguridad de aprendizaje automático del plan Plus.

Servidor de autorización administrado, servidor de autorización de interfaz de usuario alojada, servidor de autorización

Un componente del inicio de sesión gestionado que aloja los servicios para la interacción con las aplicaciones IdPs y las aplicaciones del dominio del grupo de usuarios. La interfaz de usuario administrada se diferencia del inicio de sesión administrado en las características interactivas para el usuario que ofrece, pero tiene las mismas capacidades de servidor de autorización.

Inicio de sesión administrado, páginas de inicio de sesión administrado

Conjunto de páginas web del dominio del grupo de usuarios que alojan servicios para la autenticación de usuarios. Estos servicios incluyen funciones para operar como un IdP, una parte de confianza para un tercero y un servidor de una IdPs interfaz de usuario de autenticación interactiva para el usuario. Al establecer un dominio para su grupo de usuarios, Amazon Cognito pone en línea todas las páginas del inicio de sesión administrado.

Su aplicación importa bibliotecas OIDC que invocan los navegadores de los usuarios y las dirige a la interfaz de usuario de inicio de sesión administrado para realizar las operaciones de registro, inicio de sesión, administración de contraseñas y otras operaciones de autenticación. Tras la autenticación, las bibliotecas OIDC pueden procesar el resultado de la solicitud de autenticación.

Autenticación del inicio de sesión administrado

Sign-in con los servicios del dominio de su grupo de usuarios, realizados con páginas del navegador interactivas para el usuario o solicitudes de la API HTTPS. Las aplicaciones gestionan la autenticación del inicio de sesión administrado con las bibliotecas OpenID Connect (OIDC). Este proceso incluye el inicio de sesión con proveedores externos, el inicio de sesión de usuarios locales con páginas de inicio de sesión administrado interactivas y la autorización M2M. La autenticación con la interfaz de usuario alojada clásica también se incluye en este término.

Puede comparar esto con la autenticación con AWS SDK.

Plan Plus

El plan de características con las últimas novedades y características de seguridad avanzadas en grupos de usuarios.

Autenticación de SDK, autenticación AWS de SDK

Un conjunto de operaciones de API de autenticación y autorización que puedes añadir al back-end de tu aplicación con un AWS SDK. Este modelo de autenticación necesita su propio mecanismo de inicio de sesión personalizado. La API puede iniciar sesión tanto en usuarios locales como en usuarios vinculados.

Puede comparar esto con la autenticación de inicio de sesión administrado.

Protección contra amenazas, características de seguridad avanzadas

En los grupos de usuarios, la protección contra amenazas se refiere a las tecnologías diseñadas para mitigar las amenazas a los mecanismos de autenticación y autorización. La autenticación flexible, la detección de credenciales comprometidas y las listas de direcciones IP bloqueadas pertenecen a la categoría de protección contra amenazas.

Personalización del token

Resultado de un desencadenador de Lambda previo a la generación del token que modifica el ID de usuario o el token de acceso en tiempo de ejecución.

Grupo de usuarios, proveedor de identidades 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.

Verificación

Proceso de confirmación de 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 dirección de correo electrónico o un número de teléfono nuevos. Cuando el usuario envía el código a Amazon Cognito, demuestra que es propietario del destino del mensaje y puede recibir mensajes adicionales del grupo de usuarios. Consulte también confirmación.

Perfil de usuario, cuenta de usuario

Entrada de un usuario en el directorio de usuarios. Todos los usuarios, incluidos los de terceros IdPs, tienen un perfil en su grupo de usuarios.

Grupos de identidades

Cuando vea los términos de la lista siguiente en esta guía, se refieren a una característica 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 del usuario como etiquetas a las credenciales de los usuarios.

Autenticación básica (clásica)

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

Identidades autenticadas por el desarrollador

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

Credenciales de desarrollador

Claves de la API de IAM de un administrador de grupos de identidades.

Autenticación mejorada

Flujo de autenticación que selecciona un rol de IAM y aplica las etiquetas de entidades principales de acuerdo con la lógica que defina en su grupo de identidades.

Identidad

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 AWS credenciales temporales.

Identidad no autenticada

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.

Introducción al AWS

Antes de empezar a trabajar con Amazon Cognito, prepárese con algunos recursos necesarios AWS . Si ya puede iniciar sesión en una Cuenta de AWS, puede saltarse esta sección. Siga leyendo si busca información sobre cómo registrarse e iniciar sesión con AWS credenciales. Una vez que

tenga credenciales con permisos AWS Identity and Access Management (de IAM) suficientes, podrá empezar con los grupos de usuarios y los grupos de identidades.

Regístrese para obtener una Cuenta de AWS

Para empezar AWS, necesitas un Cuenta de AWS. Hay dos formas de inscribirse en AWS. Si desea tener un control total sobre la configuración de la cuenta y el lugar donde se almacena su contenido, o si tiene cargas de trabajo reguladas, como HIPAA o FedRAMP, utilice Sign up for (avanzado). AWS Si es un usuario nuevo y desea utilizar un nombre de usuario que ya posee para iniciar sesión AWS y crear un AWS entorno con los valores predeterminados preconfigurados en Select, utilice Sign up for. Regiones de AWSAWS

Introducción a Amazon Cognito

En este capítulo, empezará con la autenticación de grupos de usuarios, es decir, cómo crear su primer grupo de usuarios e iniciar sesión con su primer usuario. Amazon Cognito también ofrece grupos de identidades, que intercambian la identidad de un usuario por AWS credenciales temporales; si eso es lo que necesita, consulte. Intercambio de identidades por credenciales AWS Si no está seguro del componente que necesita, consultethe section called "¿Qué partes de Amazon Cognito necesita?".

> **Note**
>
> Necesita una AWS cuenta para completar el tutorial y los inicios rápidos de este capítulo. Si no dispone de una, consulte the section called "Introducción al AWS".

Tiene una aplicación que requiere autenticación y control de acceso. Puede trabajar dentro del marco OpenID Connect (OIDC) para el inicio de sesión único (SSO). Amazon Cognito cuenta con herramientas para gestionar la lógica de autenticación en el back-end de la aplicación con un AWS SDK y para invocar un navegador en el cliente para acceder a un servidor de autorización gestionado.

La consola de Amazon Cognito lo guía a través de la creación de un grupo de usuarios desde el punto de vista de su marco de aplicación preferido. A partir de ahí, puede continuar añadiendo funciones como el inicio de sesión federado con redes sociales externas o proveedores de identidad de SAML 2.0 (). IdPs Los modelos de aplicaciones de la consola de Amazon Cognito se basan en la adición de bibliotecas OIDC al proyecto y en la invocación de un navegador.

A medida que vaya ampliando el 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.

Los ejemplos de este capítulo y de la consola de Amazon Cognito muestran una integración básica de los recursos de la aplicación con los grupos de usuarios de Amazon Cognito. Más adelante, podrá ajustar el grupo de usuarios para utilizar más opciones que tenga a su disposición. Luego, puedes actualizar tu aplicación para adoptar nuevas funciones e interactuar con ellas. IdPs

Si no desea utilizar las páginas de inicio de sesión administradas, puede crear una aplicación con interfaces de autenticación personalizadas mediante un AWS SDK o. AWS Amplify Las aplicaciones

que cree de esta manera interactúan con la API de grupos de usuarios y solo son adecuadas para autenticar a usuarios locales. Siga aprendiendo sobre este modelo de autenticación en Otras opciones de aplicación.

Tu primer inicio de sesión: un recorrido completo

Esta página es un mapa, no un sustituto de los procedimientos detallados a los que enlaza. Muestra cómo se conectan los pasos de configuración para que pueda ver todo el recorrido antes de sumergirse en un solo paso. Cada paso numerado enlaza con el procedimiento autorizado.

> **Note**
>
> En este tutorial se utiliza un grupo de usuarios con inicio de sesión gestionado, el punto de partida más común para una aplicación que registra usuarios. El inicio de sesión gestionado es la actual interfaz de usuario de Cognito-hosted inicio de sesión de Amazon, con un editor visual de marca; la antigua interfaz de usuario alojada clásica es una alternativa más sencilla. Tú eliges una u otra como versión de marca en el dominio de tu grupo de usuarios, y esa opción se aplica a todos los clientes de aplicaciones que utilizan el dominio. El inicio de sesión gestionado requiere el plan de funciones Essentials o Plus; en el plan Lite, la interfaz de usuario alojada clásica es la única opción alojada. Para conocer las diferencias, consulte the section called “Inicio de sesión administrado” y. the section called “Planes de características de grupo de usuarios” Si su objetivo es crear AWS credenciales temporales para una fuente de identidad existente, empiece porthe section called “Introducción a los grupos de identidades”.

De una cuenta vacía a un token validado

1. Cree un grupo de usuarios. Un grupo de usuarios es su directorio de usuarios y el proveedor de identidades que emite los tokens. Consulte the section called “Su primera aplicación y grupo de usuarios”.
2. Agrega un dominio. Las páginas de inicio de sesión gestionadas se proporcionan desde un dominio de grupo de usuarios, ya sea un dominio con prefijo de Amazon Cognito o su propio dominio personalizado. Sin un dominio, no hay una página de inicio de sesión alojada a la que enviar a los usuarios. Consulte the section called “Configuración de un dominio”.
3. Crea un cliente de aplicación. Un cliente de aplicaciones representa su aplicación ante el grupo de usuarios. Incluye la configuración que controla el inicio de sesión, como las URL de

devolución de llamada permitidas y los métodos de inicio de sesión. La versión de marca (inicio de sesión gestionado o interfaz de usuario alojada clásica) se establece en el dominio, no en el cliente de la aplicación. Consulte the section called “Clientes de aplicaciones”.

4. Envía al usuario al inicio de sesión gestionado para iniciar sesión. Con un dominio y un cliente de aplicación instalados, tu aplicación redirige a los usuarios a las páginas de inicio de sesión gestionadas para que se registren e inicien sesión. Amazon Cognito devuelve al usuario a la URL de devolución de llamada con un código de autorización. Consulte the section called “Inicio de sesión administrado”.
5. Cambie el código por fichas. La aplicación intercambia el código de autorización en el punto final del token por un token de identificación, un token de acceso y un token de actualización. Consulte the section called “Punto de conexión de token”.
6. Valide el token. Tu aplicación (o tu API) verifica la firma del token y la reclama antes de confiar en él. Consulte the section called “Verificación de un token web JSON”.

En este punto, tienes un usuario que ha iniciado sesión y un token validado. Para autorizar ese token en tu propia API o back-end (el siguiente paso más común), consulta. the section called “Acceso a recursos de ” Si, por el contrario, necesitas conceder al usuario AWS credenciales temporales para llamar a AWS servicios como Amazon S3 directamente desde tu aplicación, continúa haciéndolothe section called “Grupos de identidades de Amazon Cognito”.

Creación de una nueva aplicación en la consola de Amazon Cognito

Los grupos de usuarios añaden opciones de autenticación a las aplicaciones de software. Para empezar a usarlos de forma más sencilla, entre en la consola de Amazon Cognito y siga las instrucciones que aparecen allí. El proceso de creación que se realiza allí lo guía no solo en la configuración de los recursos del grupo de usuarios, sino también en la configuración de las partes iniciales de la aplicación.

Cuando esté listo para empezar, vaya a la consola de Amazon Cognito y seleccione el botón para crear un nuevo grupo de usuarios. El proceso de configuración lo guía a través de las opciones de configuración y lenguaje de programación.

Recursos adicionales para conceptos de autenticación

- Autenticación con grupos de usuarios de Amazon Cognito

- Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado
- Funcionamiento de la autenticación con Amazon Cognito
- Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles

Cómo crear recursos de Amazon Cognito para su aplicación

1. Vaya a la consola de Amazon Cognito. A fin de asignar permisos a la entidad principal de IAM para que pueda crear y gestionar los recursos de Amazon Cognito, consulte AWS políticas gestionadas para Amazon Cognito. La política de `AmazonCognitoPowerUser` es suficiente para crear grupos de usuarios.
2. Seleccione Crear grupo de usuarios en el menú Grupos de usuarios o seleccione Comenzar de forma gratuita en menos de cinco minutos.
3. En Defina su aplicación, elija el tipo de aplicación que mejor se adapte al escenario de aplicación para el que desea crear los servicios de autenticación y autorización.
4. En Poner un nombre a la aplicación, introduzca un nombre descriptivo o continúe con el nombre predeterminado.
5. Debe realizar algunas elecciones básicas en Configurar opciones, con ajustes que no podrá cambiar después de crear el grupo de usuarios.

    a. En Opciones para los identificadores de inicio de sesión, díganos cómo quiere identificar a los usuarios cuando inicien sesión. Puede elegir nombres de usuario, direcciones de correo electrónico o números de teléfono generados por usuarios. También puede permitir una combinación de varias opciones. Amazon Cognito acepta las opciones que se configuran aquí en el campo de nombre de usuario de los formularios de inicio de sesión con el inicio de sesión administrado.

    b. En Atributos necesarios para el inicio de sesión, díganos qué información de usuario desea recopilar cuando los usuarios registren una cuenta nueva. En las páginas del inicio de sesión administrado, Amazon Cognito presenta solicitudes para todos los atributos necesarios.

    Las opciones para los identificadores de inicio de sesión influyen en los atributos necesarios. El nombre de usuario requiere atributos de correo electrónico o teléfono para que cada usuario pueda recibir un código de restablecimiento de contraseña en un correo electrónico o mensaje SMS. El correo electrónico requiere el atributo de correo electrónico y el número de teléfono requiere el atributo de número de teléfono.

6. En Agregar una URL de retorno, introduzca una ruta de redireccionamiento a su aplicación para que los usuarios completen la autenticación. Esta ubicación debe ser una ruta de la aplicación que utilice las bibliotecas OpenID Connect (OIDC) para procesar los resultados de la autenticación de los usuarios. Un ejemplo de URL de retorno para una aplicación de prueba es `https://localhost:3000/callback`. En la aplicación NodeJS de ejemplo de la consola de Amazon Cognito, esta ruta emplea openid-client para recopilar el token de acceso y canjearlo por información del usuario. Podrás explorar ejemplos para su plataforma de desarrollo después de crear los recursos.
7. Seleccione Cree su aplicación. Amazon Cognito crea un grupo de usuarios y un cliente de aplicación con la configuración predeterminada para el tipo de aplicación. Puede configurar opciones adicionales, como proveedores de identidad externos y autenticación multifactor (MFA), después de crear los recursos iniciales.
8. En la página Configurar la aplicación, puede obtener inmediatamente ejemplos de código para su aplicación. Para explorar su nuevo grupo de usuarios, desplácese hacia abajo y seleccione Ir a descripción general.
9. Para añadir más aplicaciones al mismo grupo de usuarios, navegue hasta el menú Clientes de aplicación y añada un nuevo cliente de aplicación. De este modo, se repite el proceso de creación centrado en las aplicaciones, pero solo se añade un nuevo cliente de aplicaciones al grupo de usuarios existente.

Tras crear un grupo de usuarios y uno o más clientes de aplicación mediante este proceso, puede empezar a probar las operaciones de autenticación con el inicio de sesión administrado. Estas opciones de inicio rápido están abiertas al autorregistro público. Le recomendamos que cree un entorno de pruebas con el proceso de la consola y que, luego, pase el diseño final a producción. Dedique tiempo a familiarizarse con las funciones de Amazon Cognito. Luego, para pasar a las cargas de trabajo de producción, cree configuraciones personalizadas e impleméntelas con herramientas de automatización como AWS CloudFormation y la. Kit de desarrollo de la nube de AWS (AWS CDK)

Amazon Cognito establece algunas configuraciones predeterminadas en este proceso que no se pueden revertir. Para obtener más información sobre los ajustes del grupo de usuarios que no se podrán cambiar y las opciones que puede elegir en la consola, consulte Actualización de la configuración del grupo de usuarios y del cliente de aplicación.

Opción	Efecto	Cómo cambiar	Más información
Secreto del cliente	Requiere un hash de secreto de cliente en las solicitudes de autenticación.	Cree un nuevo cliente de aplicaciones con una aplicación web tradicional o un perfil de Machine-to-machine aplicación.	Application-specific configuración con clientes de aplicaciones
Nombre de usuario preferido	El grupo de usuarios no acepta el atributo `preferred_username` como alias.	Cree un grupo de usuarios mediante programación con un AWS SDK.	Personalización de los atributos de inicio de sesión
Sensible a mayúsculas y minúsculas	Los nombres de usuario del grupo de usuarios no distinguen entre mayúsculas y minúsculas; por ejemplo, `JohnD` se considera el mismo usuario que `johnd`.	Cree un grupo de usuarios mediante programación con un SDK. AWS	Sensibilidad de mayúsculas y minúsculas en el grupo de usuarios

Otras opciones de aplicación

Es posible que ya tenga una interfaz de usuario de aplicación existente que desee integrar con la autenticación de Amazon Cognito. Es posible que incluso tenga sus propias páginas de autenticación existentes con una configuración de directorio menos funcional que la de los grupos de usuarios de Amazon Cognito. Puede añadir o sustituir un componente de autenticación a una aplicación de este tipo con las integraciones de Amazon Cognito en AWS los SDK para diversos lenguajes de programación. A continuación se muestran algunos ejemplos.

Si crea un grupo de usuarios para este fin en la consola de Amazon Cognito, puede que no sea necesario tener un dominio de grupo de usuarios que aloje páginas de inicio de sesión interactivas y servicios de OpenID Connect (OIDC). El proceso de creación del grupo de usuarios en la consola

genera automáticamente un dominio para usted. Puede eliminar este dominio desde la pestaña Dominio de su grupo de usuarios. Otras opciones incluyen la creación programática de los recursos de Amazon Cognito para su aplicación con solicitudes de API AWS en los SDK y con las opciones de configuración automática en la CLI. AWS Amplify Para obtener más información, consulte Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles.

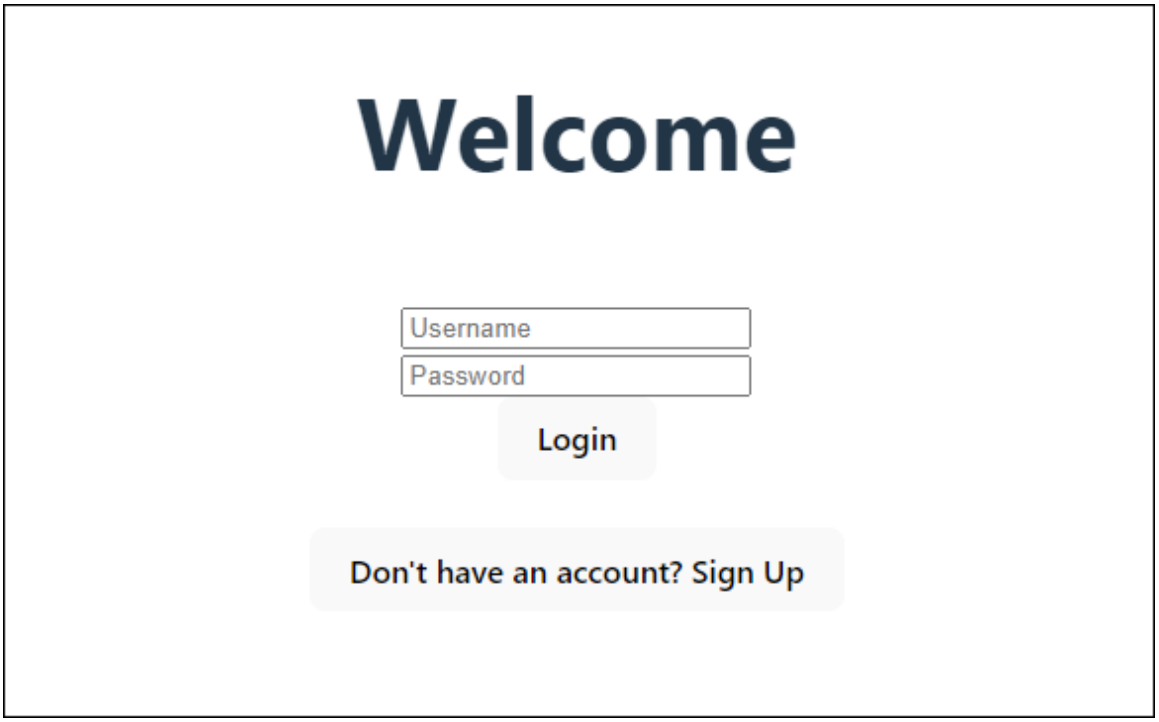
Temas

- Configuración de una aplicación de ejemplo de una sola página de React
- Configuración de una aplicación de ejemplo Android con Flutter

Configuración de una aplicación de ejemplo de una sola página de React

En este tutorial, creará una aplicación React de una sola página en la que podrá probar el registro, la confirmación y el inicio de sesión de los usuarios. React es una JavaScript-based biblioteca para 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 tienes experiencia en el desarrollo de aplicaciones web con React, descarga la aplicación de ejemplo de GitHub.

La siguiente captura de pantalla es la página de autenticación inicial de la aplicación que va a crear.



Para configurar esta aplicación, su grupo de usuarios debe cumplir 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 entre mayúsculas y minúsculas. Requisitos de nombre de usuario: la opción Establecer que el nombre de usuario distinga entre mayúsculas y minúsculas no debe estar seleccionada.
- Multi-factor no se requiere autenticación (MFA). Cumplimiento de MFA: MFA opcional.
- El grupo de usuarios verifica los atributos para la confirmación del perfil de usuario mediante un mensaje de correo electrónico. Atributos a verificar: Enviar mensaje de correo electrónico, verificar la dirección de correo electrónico.
- El correo electrónico es el único atributo obligatorio. Atributos obligatorios: correo electrónico.
- Los usuarios pueden registrarse ellos mismos en el grupo de usuarios. Self-registration: está seleccionada la opción Habilitar el registro automático.
- El cliente de 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`.

Creación de una aplicación de

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.

Creación de una aplicación web React de ejemplo

1. Inicie sesión en su entorno de desarrollador y desplácese hasta el directorio principal de su aplicación.

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

2. Cree 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/
```

4. Vaya al directorio `src` de su proyecto.

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

5. Edite `config.json` y reemplace los siguientes valores:

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

   b. Sustituya `YOUR_COGNITO_USER_POOL_ID` por el ID del grupo de usuarios que ha designado 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. Sustituya `YOUR_COGNITO_APP_CLIENT_ID` por el ID del cliente de aplicación que ha designado para las pruebas. Por ejemplo: `1example23456789`. El cliente de aplicación debe estar en el grupo de usuarios del paso anterior.

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

7. Instale la aplicación.

```
npm install
```

8. Inicie la aplicación.

```
npm run dev
```

9. Acceda a la aplicación mediante 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. Tome el código de confirmación de su 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

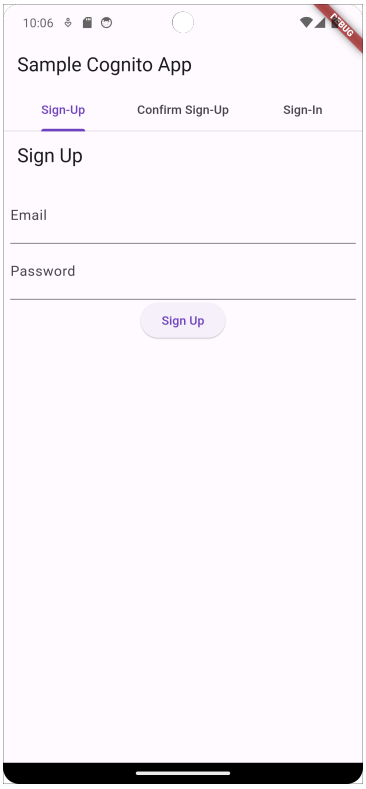
1. Vaya a la consola Lightsail. Si se le solicita, introduzca sus credenciales. AWS
2. Elija Crear instancia.
3. En Seleccione una plataforma, elija Linux/Unix.
4. En Seleccione un plano, elija Node.js.
5. En Identifique su instancia, asigne un nombre descriptivo al entorno de desarrollo.
6. Elija Crear 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 de SSH en una ventana del navegador. Ejecute `node -v` y `npm -v` para confirmar que su instancia se aprovisionó con la versión Node.js npm mínima de 10.2.3.
9. Continúe con la configuración de la aplicación React.

Configuración de una aplicación de ejemplo Android con Flutter

En este tutorial, creará una aplicación móvil en Android Studio en la que podrá 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 tienes experiencia en el desarrollo de aplicaciones móviles con Flutter, descarga la aplicación de ejemplo de. GitHub

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



Para configurar esta aplicación, su grupo de usuarios debe cumplir 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 entre mayúsculas y minúsculas. Requisitos de nombre de usuario: la opción Establecer que el nombre de usuario distinga entre mayúsculas y minúsculas no debe estar seleccionada.
- Multi-factor no se requiere autenticación (MFA). Cumplimiento de MFA: MFA opcional.
- El grupo de usuarios verifica los atributos para la confirmación del perfil de usuario mediante un mensaje de correo electrónico. Atributos a verificar: Enviar mensaje de correo electrónico, verificar la dirección de correo electrónico.
- El correo electrónico es el único atributo obligatorio. Atributos obligatorios: correo electrónico.

- Los usuarios pueden registrarse ellos mismos en el grupo de usuarios. Self-registration: está seleccionada la opción Habilitar el registro automático.
- El cliente de 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`.

Creación de una aplicación de

Creación de un ejemplo de aplicación de Android

1. Instale Android Studio y herramientas de línea de comandos.
2. En Android Studio, instale el complemento Flutter.
3. Cree un nuevo proyecto de Android Studio a partir del contenido del directorio `cognito_flutter_mobile_app` en esta aplicación de ejemplo.
   - Edite `assets/config.json` y reemplace `<<YOUR USER POOL ID>>` y `<< YOUR CLIENT ID>>` por los ID del grupo de usuarios y del cliente de aplicación.
4. Instale Flutter.
   a. Agregue Flutter a la variable PATH.
   b. Acepte licencias con el siguiente comando.

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

   c. Verifique el entorno de Flutter e instale los componentes que falten.

      ```
      flutter doctor
      ```

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

   d. Cambie al directorio del nuevo proyecto de Flutter e instale las dependencias.

      - Ejecute `flutter pub add amazon_cognito_identity_dart_2`.

   e. Ejecute `flutter pub add flutter_secure_storage`.
5. Cree un dispositivo Android virtual.
   1. En la GUI de Android Studio, cree un nuevo dispositivo con el administrador de dispositivos.
   2. En la CLI, ejecute `flutter emulators --create --name android-device`.

6. Inicie el dispositivo Android virtual.

   1. En la GUI de Android Studio, seleccione 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. Diríjase a su dispositivo virtual que se está ejecutando en Android Studio.
9. Registre un nuevo usuario con una dirección de correo electrónico válida.
10. Tome el código de confirmación de su mensaje de correo electrónico. Introduzca el código de confirmación en la aplicación.
11. Inicie sesión con su nombre de usuario y contraseña.

Incorporación de más características y opciones de seguridad al grupo de usuarios

Una vez que haya seguido los tutoriales para completar las aplicaciones de ejemplo, puede ampliar el ámbito de implementación del grupo de usuarios. O bien, si no ha creado una aplicación de prueba, cree un nuevo grupo de usuarios según sus preferencias. Puede personalizar las características del grupo de usuarios para otras aplicaciones o añadir proveedores de identidades externos. Cuando planifique la transición para incluir grupos de usuarios de Amazon Cognito en aplicaciones de producción, puede evaluar ejemplos y tutoriales adicionales.

Si su siguiente prioridad es examinar y aplicar las opciones de seguridad de las aplicaciones en sus grupos de usuarios, consulte Prácticas recomendadas de seguridad de los grupos de usuarios de Amazon Cognito.

Amazon Cognito tiene planes de características que añaden opciones funcionales y de seguridad al optar por niveles superiores. Puede empezar con el plan Lite, añadir opciones avanzadas de

autenticación y autorización con el plan Essentials y añadir barreras de protección y razonamiento automatizado con el plan Plus. Para obtener más información, consulte Planes de características de grupo de usuarios.

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

- Aplicación de la creación de marca en las páginas de inicio de sesión administrado
- Adición de MFA a un grupo de usuarios.
- Seguridad avanzada con protección contra amenazas
- Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda
- Uso de Amazon Pinpoint para analizar grupos de usuarios

Para obtener información general sobre modelos de autorización y autenticación de Amazon Cognito, consulte Funcionamiento de la autenticación con Amazon Cognito.

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

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

Temas

- Adición del inicio de sesión de redes sociales a un grupo de usuarios
- Adición de un proveedor de identidades SAML 2.0

Adición del inicio de sesión de redes sociales a un grupo de usuarios

Ofrecer a los usuarios la posibilidad de iniciar sesión en la aplicación a través de sus proveedores de identidades públicos o de redes sociales existentes puede mejorar su experiencia de autenticación. Los grupos de usuarios de Amazon Cognito se integran con los proveedores de identidad social más populares (IdPs), como Facebook, Google, Amazon y Apple, lo que ofrece a los usuarios opciones de inicio de sesión prácticas con las que ya están familiarizados.

Al configurar el inicio de sesión mediante redes sociales, ofrece a los usuarios una alternativa a la creación de una cuenta exclusiva para su aplicación. Esto puede mejorar las tasas de conversión y hacer que el proceso de registro sea más fluido. Desde la perspectiva del usuario, puede utilizar las

credenciales de redes sociales que ya posee para autenticarse rápidamente, sin la molestia de tener que recordar otro nombre de usuario y contraseña.

La configuración de un IdP de redes sociales en su grupo de usuarios supone seguir algunos pasos clave. Debe registrar la aplicación en el proveedor de redes sociales para obtener un identificador de cliente y un secreto. A continuación, puede agregar la configuración del IdP de redes sociales a su grupo de usuarios, especificando los ámbitos que desea solicitar y los atributos del grupo de usuarios que desea asignar a partir de los atributos del IdP. En tiempo de ejecución, Amazon Cognito administra el intercambio de tokens con el proveedor, asigna los atributos de los usuarios y emite los tokens para la aplicación en el formato de grupo de usuarios compartido.

Registrarse en un proveedor de identidad social

Para poder crear un IdP de redes sociales 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 tiene una aplicación de Facebook, verá otra opción. 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 Seleccionar la plataforma, seleccione las plataformas y, a continuación, seleccione Siguiente.
10. Seleccione Save changes (Guardar cambios).
11. ParaDominios de aplicación, introduzca el dominio del grupo de usuarios.

    ```
    https://your_user_pool_domain
    ```
12. Seleccione Save changes (Guardar cambios).

13. En la barra de navegación elija Productos y, a continuación, Configurar en Iniciar sesión con Facebook.
14. En el menú Iniciar sesión con Facebook Configurar, elija Configuración.

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

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

15. Seleccione Save changes (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 Aplicaciones y servicios en la barra de navegación de la parte superior de la página y, a continuación, elija 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 de la izquierda, elija API y servicios y luego Pantalla de consentimiento de Oauth.
7. Introduzca la información de la aplicación, un dominio de aplicaciones, los dominios autorizados y la información de contacto del desarrollador. Los dominios autorizados deben incluir `amazoncognito.com` y la raíz del dominio personalizado. Por ejemplo: `example.com`. Elija SAVE AND CONTINUE (GUARDAR Y CONTINUAR).
8. 1. En Ámbitos, elija 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 GUARDAR Y CONTINUAR.
10. Expanda de nuevo la barra de navegación izquierda, elija API y servicios y luego Credenciales.
11. Elija CREAR CREDENCIALES y luego 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://<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 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 Seleccionar 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 enUso de proveedores de identidades de redes sociales con un 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. Tome nota del ID de servicios, porque necesitará este valor para configurar Apple como proveedor de identidades en el Uso de proveedores de identidades de redes sociales con un grupo de usuarios.
    3. Seleccione Continue (Continuar) y, a continuación, Register (Registrarse).
13. Elija el ID de servicios que acaba de crear en la página de identificadores.

    1. Selecciona Iniciar sesión con Apple y, a continuación, selecciona 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. Elija Siguiente y, a continuación, elija 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. Selecciona Iniciar sesión con Apple y, a continuación, selecciona Configurar.
    3. En la página Configurar clave, seleccione el ID de aplicación creado anteriormente como ID de aplicación principal. Seleccione Save.
    4. Seleccione Continue (Continuar) y, a continuación, Register (Registrarse).
17. En la página Descargar clave, elija Descargar para descargar la clave privada, tome nota del ID de la clave y, a continuación, seleccione 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 Uso de proveedores de identidades de redes sociales con un 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 Consola de administración de AWS

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. Seleccione el menú Proveedores sociales y externos. 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: escriba el ID de cliente de la aplicación y la Clave secreta de cliente de aplicación que se ha generado en la sección anterior.
    - Facebook: escriba el ID de cliente de la aplicación y la Clave secreta de cliente de aplicación que se ha 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 obsolescencia. Los ámbitos y atributos de Facebook pueden variar según las versiones de la API. Te recomendamos que pruebes el inicio de sesión con tu identidad social con Facebook para asegurarte de que la federación funciona según lo previsto.

- Inicio de sesión con Apple: escriba el ID de servicios, ID de equipo, ID de clave y la clave privada que se ha generado 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. En Iniciar sesión con Apple, selecciona las casillas de verificación de los ámbitos a los que quieres 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.

- El 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 usa en grupos de usuarios y and/or 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 ha eliminado un usuario y luego ese mismo usuario vuelve a iniciar sesión sin quitar 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 Cosas que debe saber acerca de los asignaciones.
9. Seleccione Crear.
10. En el menú Clientes de aplicación, elija uno de los clientes de aplicación en la lista y seleccione Editar la configuración de interfaz de usuario alojada. Agregue el nuevo proveedor de identidad social al cliente de aplicación en Identity providers (Proveedores de identidad).
11. Seleccione Save changes (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.auth.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 al inicio de sesión administrado. La página muestra un mensaje de error `Something went wrong`.

Adición de un proveedor de identidades SAML 2.0

Los usuarios de la aplicación pueden iniciar sesión con un proveedor de identidades (IdP) SAML 2.0. Puede elegir SAML 2.0 en lugar IdPs de redes sociales IdPs si sus clientes son clientes internos o empresas vinculadas a su organización. Si un IdP de redes sociales permite que todos los usuarios se registren en una cuenta, es más probable que un IdP SAML se vincule con un directorio de usuarios que controle su organización. 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 identidades SAML con un grupo de usuarios.

Debe actualizar el proveedor de identidades SAML y configurar su grupo de usuarios. Para obtener más información acerca de cómo agregar su grupo de usuarios como relación de confianza o aplicación para su proveedor de identidades SAML 2.0, consulte la documentación de su proveedor de identidades SAML.

También debe proporcionar un punto de conexión del servicio de consumidor de aserción (ACS) a su proveedor de identidades 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 del grupo 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
```

El prefijo del dominio y el valor de la región de su grupo de usuarios se encuentran en el menú Dominio de la consola de Amazon Cognito.

Para algunos proveedores de identidades SAML, también tiene que proporcionar el proveedor de servicios (SP) `urn`, también denominado URI de audiencia o ID de entidad del SP, con el formato:

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

Puede encontrar su ID de grupo de usuarios en el panel de control Información general de su grupo de usuarios en 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". 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 de SAML del proveedor de identidades 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 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.
4. Seleccione el menú Proveedores sociales y externos. 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 introduce un usuario al iniciar sesión. A continuación, dirige al usuario al proveedor que corresponde 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 conexión de `https://<your Amazon Cognito domain>/saml2/logout` que se crea al configurar el inicio de sesión administrado. El punto de conexión `saml2/logout` utiliza el enlace POST.

> **Note**
>
> Si se selecciona esta opción y el proveedor de identidades SAML espera una solicitud de cierre de sesión firmada, también se deberá configurar el certificado de firma que proporciona Amazon Cognito en dicho proveedor.
> El IDP de SAML procesa la solicitud de cierre de sesión firmada y cierra la sesión del usuario de la sesión 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, en vez de cargar un archivo, introduzca la URL de un documento de metadatos si el proveedor dispone de un punto de conexión público. 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 aserciones SAML de ejemplo y como referencia. Algunos proveedores de identidad utilizan nombres simples, por ejemplo`email`, mientras que otros utilizan nombres de URL-formatted atributos, como el siguiente ejemplo:

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

10. Seleccione Crear.

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.

Realización de 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.

Adición de código de aplicación a partir de ejemplos

En el capítulo de ejemplos de código de esta guía se incluye código de aplicación que se 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 de aplicaciones integrales 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 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

- Authentication flow examples with .NET for Amazon Cognito
- Amazon Cognito Passwordless Auth
- 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
- Building fine-grained authorization using Amazon Cognito, API Gateway, and IAM
- CloudFrontautorización @edge

Más talleres

- Implemente la autenticación sin contraseña con Amazon Cognito y WebAuthn
- Taller de Amazon Cognito
- Taller de resolución de problemas de Amazon Cognito

- Autenticación y autorización con Amazon Cognito y Verified Permissions
- Amazon Cognito JWT Deep Dive

Publicaciones de blog

- Proteja los clientes públicos de Amazon Cognito mediante un proxy de Amazon CloudFront
- How to set up Amazon Cognito for federated authentication using Azure AD
- Simplify web app authentication: A guide to AD FS federation with Amazon Cognito user pools

Planificación del diseño de Amazon Cognito

Es más fácil tomar las decisiones de este capítulo por adelantado que cambiarlas más adelante. En este capítulo, se tratan los siguientes temas:

- Elegir un enfoque de autenticación: cómo inician sesión los usuarios.
- Escenarios comunes: patrones de arquitectura que se adaptan a los de su aplicación.
- Multi-tenancy—si presta servicios a más de una organización de clientes, se describen las estrategias para aislar a los usuarios y los datos de cada inquilino. Creación de aplicaciones B2B multiusuario

Elija su método de autenticación

Al autenticar a los usuarios con un grupo de usuarios de Amazon Cognito, usted elige cómo se presenta la experiencia de inicio de sesión. Los tres enfoques siguientes difieren en la cantidad de interfaz que proporciona Amazon Cognito y en la cantidad que se crea, y no se excluyen mutuamente: puede combinarlos. Todos se autentican con el mismo grupo de usuarios y producen los mismos tokens. La diferencia es dónde se encuentran las páginas de inicio de sesión y cuánto control tienes.

Inicio de sesión gestionado (Amazon Cognito aloja las páginas)

> Amazon Cognito ofrece las páginas de registro, inicio de sesión, autenticación multifactorial y restablecimiento de contraseñas del dominio de su grupo de usuarios. Puede personalizar su aspecto con el editor de marca. Elija esta opción cuando desee una experiencia de inicio de sesión completa y alojada sin crear una interfaz de usuario y cuando desee que los flujos de OAuth 2.0 y OpenID Connect basados en estándares se gestionen por usted. El inicio de sesión gestionado es la versión actual de estas Cognito-hosted páginas de Amazon; la antigua interfaz de usuario alojada clásica es una alternativa más sencilla y de primera generación que aún puedes seleccionar por cliente de aplicación. Consulte the section called “Inicio de sesión administrado”.

API de grupos de usuarios (tú construyes la interfaz)

> La aplicación recopila las credenciales y llama a la API de grupos de usuarios de Amazon Cognito directamente a través de un AWS SDK. Usted crea y controla toda la interfaz. Elija esta opción cuando necesite una experiencia de inicio de sesión totalmente personalizada o

cuando inicie sesión para los usuarios desde un servicio de back-end. Consulte the section called "Autenticación".

Third-party proveedores de identidad (federación)

Sus usuarios inician sesión a través de un proveedor externo. Puede ser un proveedor social como Google o Apple, o un proveedor empresarial de SAML u OIDC. Amazon Cognito federa su identidad en su grupo de usuarios. El inicio de sesión federado se realiza a través de los puntos de acceso alojados en el dominio del grupo de usuarios, ya sea el inicio de sesión administrado o los puntos de enlace de autorización e inicio de sesión a los que se redirige desde una interfaz personalizada. La operación de la API de grupos de usuarios nativos `InitiateAuth` no permite que un usuario inicie sesión a través de un proveedor federado, por lo que la federación siempre utiliza los puntos de enlace alojados en lugar del flujo de autenticación directa del SDK. Elija esta opción cuando sus usuarios ya tengan identidades en otros lugares. Consulte the section called "Third-party Inicio de sesión con IdP".

En la siguiente tabla se resume cuándo utilizar cada enfoque.

Si desea...	Comience con
Envíe el inicio de sesión rápidamente sin necesidad de crear y mantener una interfaz de usuario	Inicio de sesión administrado
Controla completamente el aspecto y el comportamiento de todas las pantallas de inicio de sesión	La API de grupos de usuarios
Autentica las solicitudes de una persona de back-end o de una persona que llama de máquina a máquina	La API de grupos de usuarios (y, en el caso de M2M, las credenciales de los clientes)
Permite a los usuarios iniciar sesión con Google, Apple o un directorio empresarial	Third-party proveedores de identidad (federación), mediante el inicio de sesión gestionado o los puntos finales de autorización e inicio de sesión alojados

> **Note**
>
> Sea cual sea el enfoque que elija, el plan de funciones de su grupo de usuarios determina qué capacidades están disponibles. El plan Lite cubre el registro y el inicio de sesión básicos; Essentials añade opciones como el inicio de sesión sin contraseña y con clave de paso y el inicio de sesión gestionado; y Plus añade una protección avanzada contra amenazas. Usted selecciona un plan al crear un grupo de usuarios y puede cambiarlo más adelante, por lo que no tiene que decidirlo antes de empezar. Si una función que esperas no está disponible, consulta primero tu plan. Para ver una comparación completa, consultathe section called "Planes de características de grupo de usuarios".

> **Note**
>
> Todos estos enfoques responden a la pregunta ¿quién es este usuario? (autenticación). Si también necesitas conceder a los usuarios AWS credenciales temporales para acceder a servicios como Amazon S3, los grupos de identidades responden a otra pregunta (autorización). Consulte the section called "¿Qué partes de Amazon Cognito necesita?".

Situaciones comunes de Amazon Cognito

En este tema se describen siete escenarios comunes para usar 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 que sus usuarios puedan acceder a otros Servicios de AWS.

Un grupo de usuarios es un directorio de usuarios en Amazon Cognito. Los usuarios de la aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidades (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

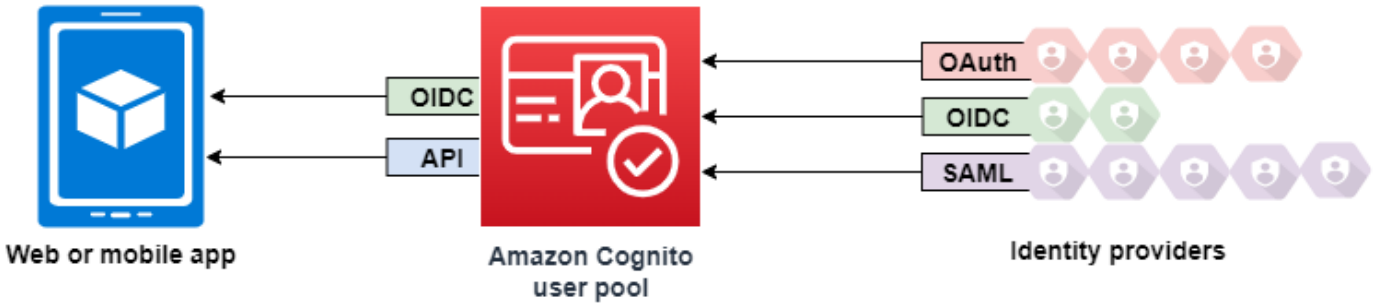
- Autenticar con un grupo de usuarios
- Acceso a los recursos de backend con tokens de grupos de usuarios
- Acceso a los recursos con API Gateway y Lambda mediante un grupo de usuarios
- Acceso AWS servicios con un grupo de usuarios y un grupo de identidades
- Autenticate con un tercero y accede AWS servicios con un grupo de identidades
- Acceso AWS AppSync recursos con Amazon Cognito
- Autoriza aplicaciones de máquina a máquina (M2M)
- Step-up authentication

Autenticar con un grupo de usuarios

Puede permitir que los usuarios se autentiquen con un grupo de usuarios. Los usuarios de la aplicación pueden iniciar sesión directamente a través del grupo de usuarios o pueden federarse a través de un proveedor de identidades (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

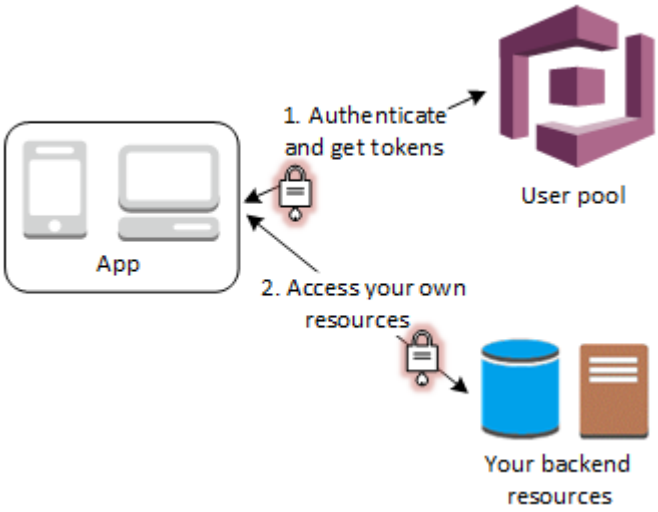
Tras una autenticación correcta, su aplicación web o móvil recibe los tokens del grupo de usuarios de Amazon Cognito. Puede usar esos tokens para recuperar AWS las credenciales que permiten a su aplicación acceder a otros AWS servicios, o puede optar por usarlos para controlar el acceso a los recursos del lado del servidor o a Amazon API Gateway.

Para obtener más información, consulte Un ejemplo de sesión de autenticación y Descripción de los tokens web JSON (JWT) de los grupos de usuarios.



Acceso a los recursos de backend con tokens de grupos de usuarios

Tras iniciar sesión correctamente en el grupo de usuarios, su aplicación web o móvil recibirá los tokens del grupo de usuarios de 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 Ámbitos, M2M y servidores de recursos.

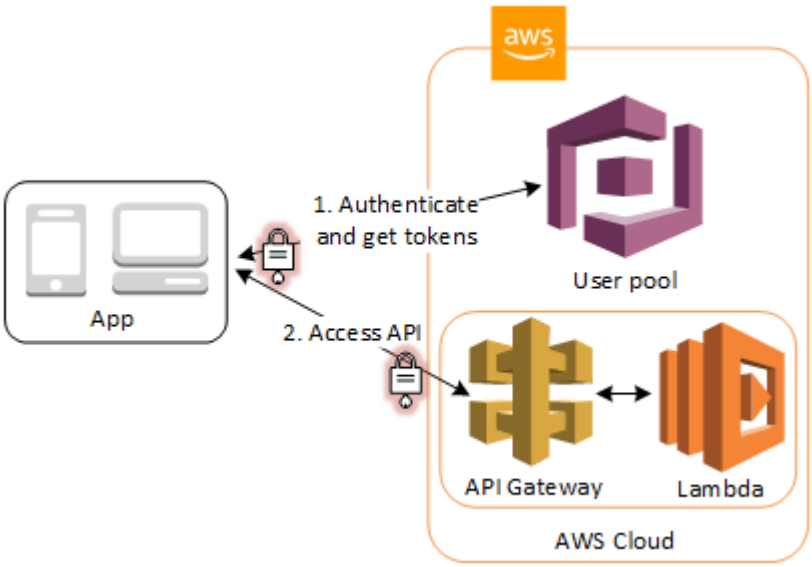
Para obtener más información sobre la autenticación de grupos de usuarios, consulte Un ejemplo de sesión de autenticación y Descripción de los tokens web JSON (JWT) de los grupos de usuarios.

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.

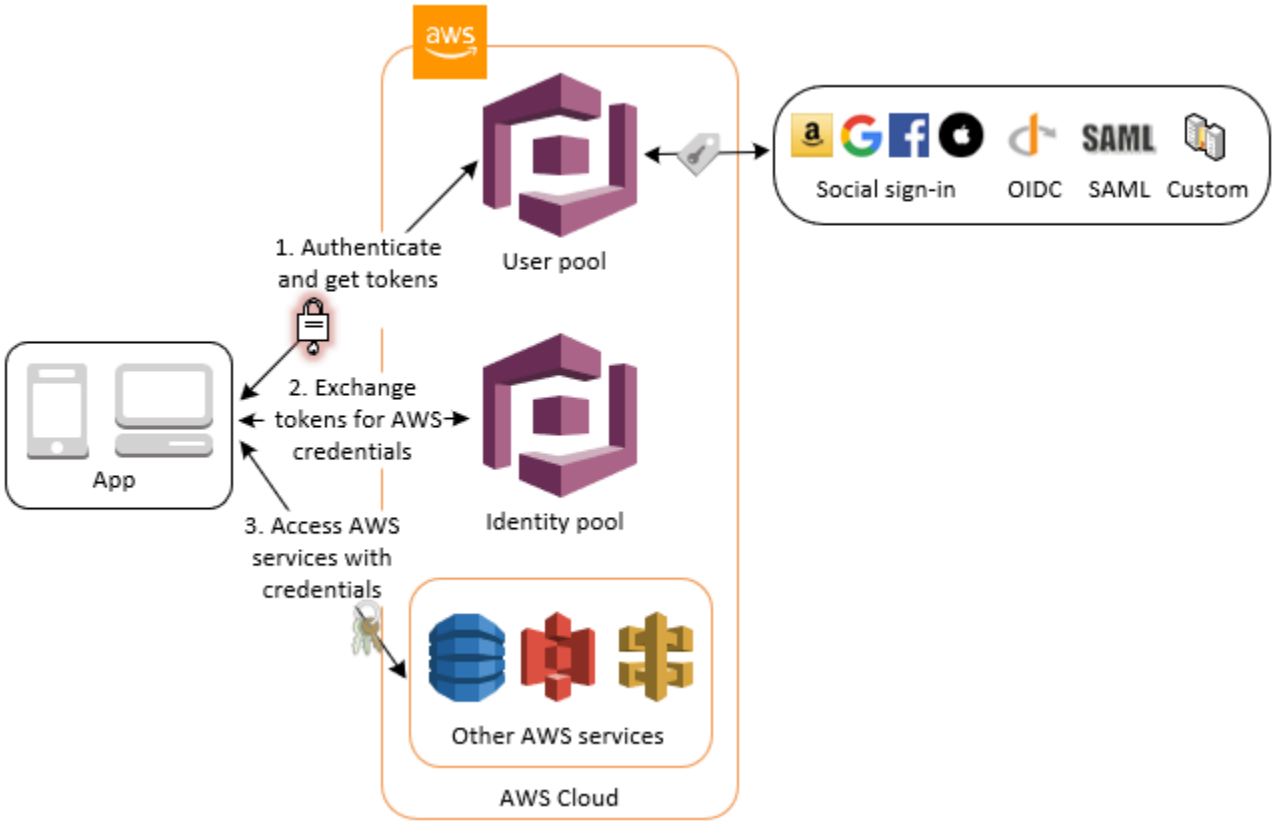
Puede utilizar grupos en un grupo de usuarios a fin de controlar permisos con API Gateway al mapear membresías de grupo a roles de IAM. Los grupos a los que pertenece un usuario están incluidos en el token de ID proporcionado por un grupo de usuarios cuando el usuario de la aplicación inicia sesión. Para obtener más información sobre los grupos 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.



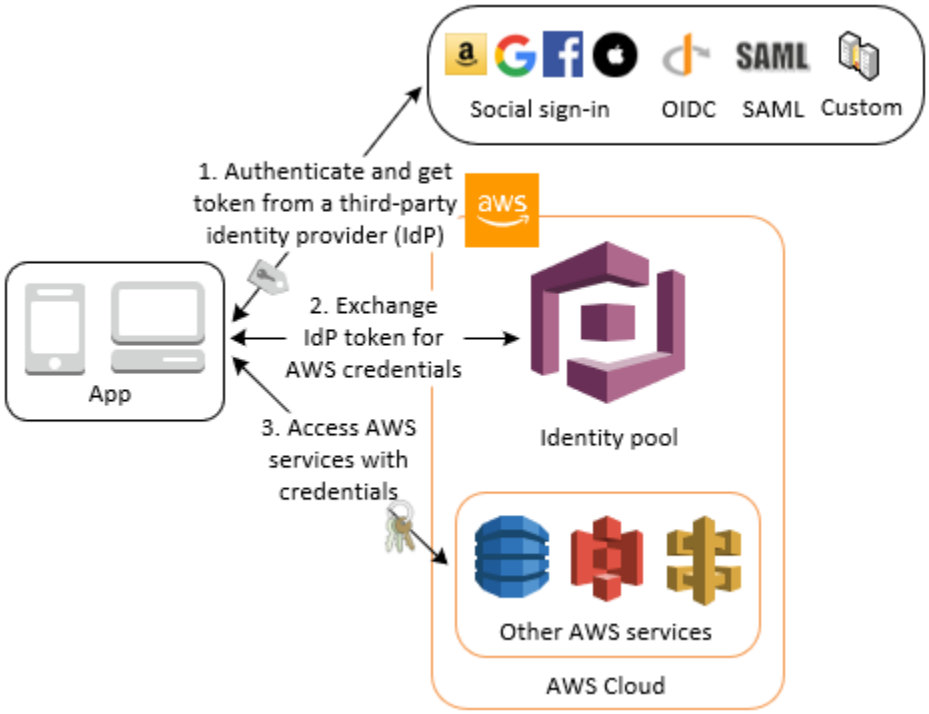
Acceso AWS servicios con un grupo de usuarios y un grupo de identidades

Tras una autenticación correcta del grupo de usuarios, la aplicación recibe los tokens del grupo de usuarios de Amazon Cognito. Puede cambiarlos por un acceso temporal a otros AWS servicios con un grupo de identidades. Para obtener más información, consulte Acceso Servicios de AWS usar un grupo de identidades después de iniciar sesión y Introducción a los grupos de identidades de Amazon Cognito.



Autenticate con un tercero y accede AWS servicios con un grupo de identidades

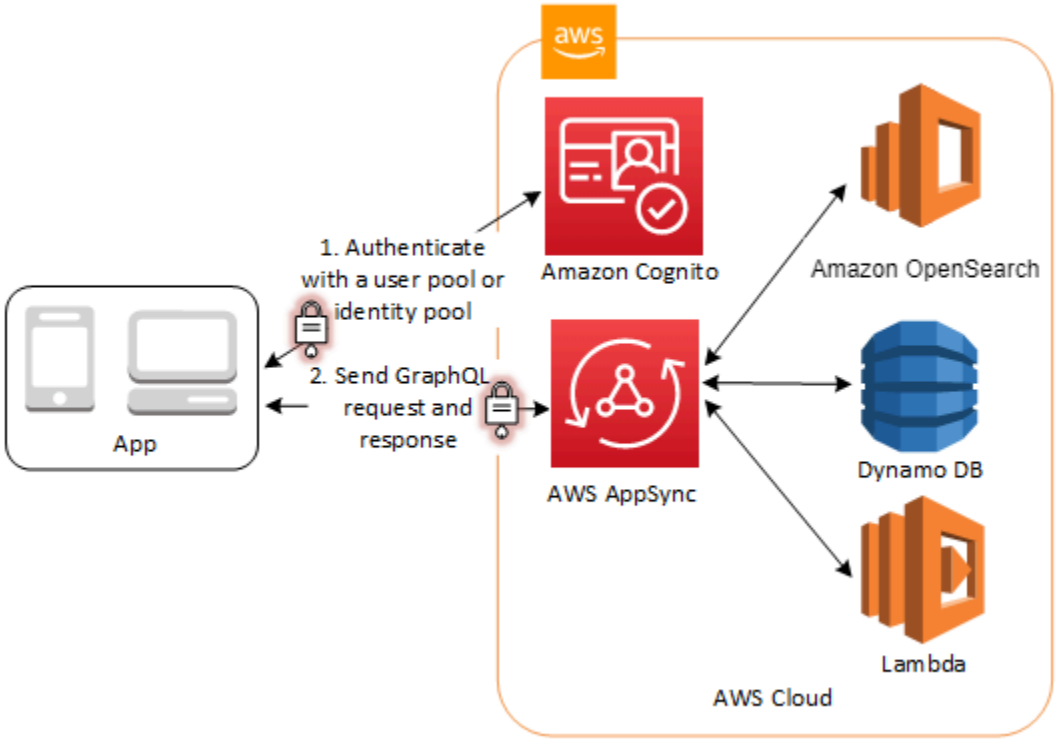
Puede permitir que sus usuarios accedan a los AWS servicios a través de un grupo de identidades. Un grupo de identidades requiere un token de proveedor de identidad de un usuario que se haya autenticado mediante un proveedor de identidad de terceros (o nada si se trata de un invitado anónimo). A cambio, 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.



Acceso AWS AppSync 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.



Autoriza aplicaciones de máquina a máquina (M2M)

Con la autorización de máquina a máquina (M2M), una aplicación, servicio o dispositivo puede autenticarse por sí mismo y obtener acceso autorizado a sus API, sin necesidad de que un usuario inicie sesión. Amazon Cognito autoriza una aplicación en lugar de un usuario y solo emite un token de acceso, sin un token de ID ni un token de actualización. Este es el modelo para cargas de trabajo, como una aplicación que recopila datos de forma autónoma de una flota de dispositivos de Internet de las cosas (IoT), donde cada dispositivo se conecta a su API y envía los datos sin intervención humana.

Para habilitar la autorización M2M, cree un cliente de aplicación que tenga un secreto de cliente y actívelo para la concesión. `client_credentials` La identidad de su máquina solicita los tokens del servidor de autorización y los presenta al servidor de recursos. Para obtener más información, consulte Machine-to-machine autorización (M2M) y Punto de conexión de token.

Step-up authentication

Considere una aplicación bancaria que permita a los usuarios iniciar sesión con un nombre de usuario y una contraseña. Para realizar una transferencia bancaria, la aplicación requiere una

autenticación multifactor (MFA). Cuando un usuario intenta realizar una transferencia, la aplicación inspecciona la `acr` afirmación en los tokens de acceso e ID del usuario, determina que el nivel de autenticación es insuficiente y redirige al usuario a Amazon Cognito para completar la MFA. A continuación, Amazon Cognito emite un nuevo token con un nivel de autenticación elevado y la aplicación permite la transferencia.

Antes de redirigir a un usuario a Amazon Cognito para una autenticación avanzada, evalúe los tokens que ya contiene la aplicación. Compare la `auth_time` afirmación con la antigüedad máxima que permite la operación y compare el `acr` nivel del token con el nivel que requiere la operación. Redirija al usuario a Amazon Cognito para completar el flujo de autenticación solo cuando aún no se haya alcanzado el nivel de autenticación o la actualización requeridos.

Para obtener más información sobre los niveles de ACR, los valores de AMR y otros conceptos de autenticación avanzada, consulte. the section called “Niveles de autenticación con notificaciones de ACR y AMR” Para conocer la mecánica de solicitar una autenticación escalonada, consulte Step-up autenticación con el flujo USER_AUTH el `USER_AUTH` flujo Autorizar punto de conexión y Punto de conexión Login el inicio de sesión gestionado.

Configurar tu superficie de inicio de sesión

Antes de que tus usuarios puedan autenticarse, debes configurar la superficie a través de la cual inician sesión. En el caso del inicio de sesión gestionado, se trata de un dominio de grupo de usuarios para mostrar las páginas y un cliente de aplicación que dirija a ellas, además de la marca que utilices. En este capítulo, se tratan los siguientes temas:

- Inicio de sesión gestionado: las páginas de inicio de sesión alojadas, el dominio del grupo de usuarios que las sirve y el editor de marca.
- Migración desde la interfaz de usuario alojada clásica: lo que cambia cuando se mueve un grupo de usuarios existente a un inicio de sesión gestionado.

Para decidir, en primer lugar, entre el inicio de sesión gestionado, la API de grupos de usuarios y la federación, consulte. the section called “Elija un enfoque” Para ver los flujos de autenticación propiamente dichos, consultethe section called “Autenticación”.

Inicio de sesión administrado de grupos de usuarios

Puede elegir un dominio web para alojar los servicios de su grupo de usuarios. Un grupo de usuarios de Amazon Cognito obtiene las siguientes funciones al añadir un dominio, denominadas colectivamente inicio de sesión administrado.

- Un servidor de autorización que actúa como proveedor de identidades (IdP) para las aplicaciones que funcionan con OAuth 2.0 y OpenID Connect (OIDC). El servidor de autorización enruta las solicitudes, emite y administra los tokens web JSON (JWT) y proporciona información sobre los atributos del usuario.
- Una interfaz de usuario (UI) lista para usar en operaciones de autenticación como el inicio de sesión, el cierre de sesión y la administración de contraseñas. Las páginas de inicio de sesión administrado actúan como frontend web para los servicios de autenticación.
- Un proveedor de servicios (SP) o una parte de confianza (RP) para SAML 2.0, OIDC IdPs, Facebook IdPs, Login with Amazon, Sign in with Apple y Google.

El inicio de sesión gestionado tiene una alternativa: la clásica interfaz de usuario alojada. Puede elegir uno u otro para el dominio de un grupo de usuarios: el inicio de sesión gestionado y la interfaz de usuario alojada clásica no se utilizan juntos. La interfaz de usuario alojada clásica es una versión

de primera generación de los servicios del inicio de sesión administrado. Los servicios de interfaz de usuario alojada, IdP y RP generalmente tienen las mismas características que el inicio de sesión administrado, pero las páginas de inicio de sesión tienen un diseño más simple y menos características. Por ejemplo, el inicio de sesión con clave de acceso no está disponible en la interfaz de usuario alojada clásica. En el plan de características Lite, la interfaz de usuario alojada clásica es la única opción para los servicios de dominio del grupo de usuarios.

Las páginas de inicio de sesión administrado son una colección de interfaces web para las actividades básicas de registro, inicio de sesión, autenticación multifactor y restablecimiento de contraseña en el grupo de usuarios. También conectan a los usuarios con uno o más proveedores de identidad externos (IdPs) cuando se quiere ofrecer a los usuarios la opción de inicio de sesión. La aplicación puede invocar las páginas de inicio de sesión administrado en los navegadores de los usuarios cuando desee autenticar y autorizar a los usuarios.

Puede hacer que la experiencia de usuario de inicio de sesión administrado se adapte a la marca con logotipos, fondos y estilos personalizados. Tiene dos opciones de marca que puede aplicar a la interfaz de usuario del inicio de sesión administrado: el editor de marcas, para el inicio de sesión administrado, y la marca de la interfaz de usuario alojada (clásica), para la interfaz de usuario alojada.

Editor de marcas

> Una experiencia de usuario renovada con las opciones de autenticación más actualizadas y un editor visual en la consola de Amazon Cognito.

Marca de interfaz de usuario alojada (clásica)

> Una experiencia de usuario familiar para los grupos de usuarios anteriores de Amazon Cognito. La marca de la interfaz de usuario alojada es un sistema basado en archivos. Para aplicar la marca a las páginas de la interfaz de usuario alojadas, debe cargar un archivo de imagen del logotipo y un archivo que establezca los valores de varias opciones de estilo CSS predeterminadas.

El editor de marcas no está disponible en todos los planes de características para grupos de usuarios. Para obtener más información, consulte Planes de características de grupo de usuarios.

Para obtener más información sobre cómo crear solicitudes para los servicios de inicio de sesión administrado y de interfaz de usuario alojada, consulte Referencia de los puntos de conexión de grupos de usuarios y del inicio de sesión administrado.

El inicio de sesión gestionado permite aumentar la autenticación con los `max_age` parámetros `acr_values` y de los puntos finales de autorización e inicio de sesión. Para obtener más información, consulte el punto final de autorización y el punto final de inicio de sesión.

> **Note**
>
> El inicio de sesión administrado de Amazon Cognito no admite autenticación personalizada con desencadenadores de Lambda de desafíos de autenticación personalizados.

Temas

- Localización de inicio de sesión administrado
- Documentos de términos
- Configuración del inicio de sesión administrado con la consola de Amazon Cognito.
- Configurar el inicio de sesión gestionado con AWS Amplify
- Consulta de la página de inicio de sesión
- Personalización de las páginas de autenticación
- Cosas que debe saber sobre el inicio de sesión administrado y la interfaz de usuario alojada
- Configuración de un dominio del grupo de usuarios
- Aplicación de la creación de marca en las páginas de inicio de sesión administrado

Localización de inicio de sesión administrado

El inicio de sesión administrado se establece de forma predeterminada en inglés en las páginas interactivas para el usuario. Puede mostrar sus páginas de inicio de sesión administrado localizadas para el idioma que elija. Los idiomas disponibles son los que están disponibles en Consola de administración de AWS. En el enlace que distribuya a los usuarios, agregue un parámetro de consulta `lang`, como se muestra en el siguiente ejemplo.

```
https://<your domain>/oauth2/authorize?lang=es&response_type=code&client_id=<your app
 client id>&redirect_uri=<your relying-party url>
```

Amazon Cognito establece una cookie en el navegador de los usuarios con su idioma preferido después de la solicitud inicial con un parámetro `lang`. Una vez establecida la cookie, la selección de

idioma se mantiene sin que se muestre el parámetro ni se le pida que lo incluya en las solicitudes. Por ejemplo, cuando un usuario realiza una solicitud de inicio de sesión con un parámetro `lang=de`, sus páginas de inicio de sesión administrado se muestran en alemán hasta que borre las cookies o haga una nueva solicitud con un nuevo parámetro de localización, por ejemplo `lang=en`.

La localización solo está disponible para el inicio de sesión administrado. Debe tener el plan de características Essentials o Plus y haber asignado su dominio para el uso de la creación de marca en inicio de sesión administrado.

La selección que hace el usuario en el inicio de sesión administrado no está disponible para desencadenadores de envíos de correo o SMS personalizados. Al implementar estos desencadenadores, debe utilizar otros mecanismos para determinar el idioma preferido del usuario. En los flujos de inicio de sesión, el atributo `locale` puede indicar el idioma preferido del usuario en función de la ubicación. En los flujos de registro, la región o el ID de cliente de la aplicación de su grupo de usuarios pueden indicar una preferencia de idioma.

Están disponibles los siguientes idiomas.

Idiomas de inicio de sesión administrado

Idioma	Código
Alemán	de
Inglés	en
Español	es
Francés	fr
Bahasa Indonesia	id
Neerlandés	nl
Italiano	it
Japonés	ja
Coreano	ko
Portugués (Brasil)	pt-BR

Idioma	Código
Chino simplificado	zh-CN
Chino tradicional	zh-TW

Documentos de términos

Puede configurar sus páginas de inicio de sesión administrado para que muestren enlaces a sus documentos de Términos de uso y Política de privacidad cuando los usuarios se registren. Al configurar ambos documentos de condiciones en el cliente de aplicación, los usuarios ven el siguiente texto durante el registro: By signing up, you agree to our Terms of use and Privacy policy. Las frases Términos de uso y Política de privacidad aparecen en su página de inicio de sesión administrado, con un hipervínculo a sus documentos.

Los documentos de condiciones admiten direcciones URL de idiomas específicos que se ajustan a la localización del inicio de sesión administrado. Cuando los usuarios seleccionan un idioma con el parámetro de consulta `lang`, Amazon Cognito muestra enlaces a sus documentos de condiciones en ese idioma. Si no ha configurado una URL para un idioma específico, Amazon Cognito usa la URL predeterminada que configuró para el cliente de aplicación.

A fin de configurar los documentos de condiciones para su cliente de aplicación, vaya al menú Inicio de sesión administrado de su grupo de usuarios. En Documentos de términos, seleccione Crear documento de términos.

Amazon Cognito console

Cómo crear un documento de términos

1. Vaya a su grupo de usuarios y seleccione el menú Inicio de sesión administrado. Busque Documentos de términos.
2. Elija Crear documento de términos.
3. Seleccione el cliente de aplicación al que desea asignar el documento de términos.
4. Introduzca un Nombre de términos. Esto identifica el documento en la consola.
5. En Enlaces, elija un idioma e introduzca la URL en la que aloja el documento de términos en ese idioma.
6. Para añadir direcciones URL en otros idiomas, seleccione Agregar otro.

7. Seleccione Crear.

Amazon Cognito user pools API

A continuación, se muestra un ejemplo de cuerpo de la solicitud CreateTerms. Hace que la página de registro del cliente de la aplicación `1example23456789` muestre enlaces para una French-language y una versión Portuguese-language (de Brasil) de la política de privacidad cuando el inicio de sesión administrado está traducido a ese idioma. Es necesaria una solicitud independiente para configurar las URL `terms-of-use` antes de que el inicio de sesión gestionado muestre los enlaces en la página de registro.

```
{
   "ClientId": "1example23456789",
   "Enforcement": "NONE",
   "Links": {
      "cognito:default" : "https://example.com/privacy/",
      "cognito:french" : "https://example.com/fr/privacy/",
      "cognito:portuguese-brazil" : "https://example.com/pt/privacy/"
   },
   "TermsName": "privacy-policy",
   "TermsSource": "LINK",
   "UserPoolId": "us-east-1_EXAMPLE"
}
```

> **Note**
>
> Debe crear un documento de condiciones de uso y de política de privacidad para el cliente de la aplicación antes de que Amazon Cognito muestre los documentos de condiciones en las páginas de inicio de sesión gestionadas.

Configuración del inicio de sesión administrado con la consola de Amazon Cognito.

El primer requisito para el inicio de sesión administrado y la interfaz de usuario alojada es un dominio de grupo de usuarios. En la consola de grupos de usuarios, vaya a la pestaña Dominio de su grupo de usuarios y añada un dominio de Cognito o un dominio personalizado. También puede elegir un dominio durante el proceso de creación de un nuevo grupo de usuarios. Para obtener más

información, consulte Configuración de un dominio del grupo de usuarios. Cuando un dominio está activo en su grupo de usuarios, todos los clientes de la aplicación publican páginas de autenticación públicas en ese dominio.

Cuando crea o modifica un dominio de grupo de usuarios, configura la versión de marca para su dominio. Esta versión de marca es una opción entre el inicio de sesión administrado o la interfaz de usuario alojada (clásica). La versión de marca que elija se aplica a todos los clientes de aplicaciones que utilizan los servicios de inicio de sesión de su dominio.

El siguiente paso es crear un cliente de aplicación desde la pestaña Clientes de aplicación de su grupo de usuarios. Durante el proceso de creación de un cliente de aplicación, Amazon Cognito le pide información sobre la aplicación y, a continuación, le pide que seleccione una URL de retorno. La URL de retorno también se denomina URL del actor de confianza (RP), URI de redirección y URL de devolución de llamada. Esta es la URL desde la que se ejecuta la aplicación, por ejemplo, `https://www.example.com` o `myapp://example`.

La versión de marca que defina en el dominio determinará el estilo de sus páginas de inicio de sesión: elija Inicio de sesión administrado para diseñar páginas en el editor de marca o Interfaz de usuario alojada (clásica) para aplicar la marca de la interfaz de usuario alojada basada en archivos. Para obtener más información sobre el editor de marca y los planes de funciones que lo admiten, consulte. Planes de características de grupo de usuarios

Tras configurar un dominio y un cliente de aplicaciones y establecer una versión de marca en el grupo de usuarios, las páginas de inicio de sesión gestionadas estarán disponibles en Internet.

Cómo configurar el inicio de sesión administrado en la consola de Amazon Cognito.

1. Agrega un dominio de grupo de usuarios. En la consola de Amazon Cognito, elija su grupo de usuarios, abra la pestaña Dominio y añada un dominio de Cognito (dominio de prefijo) o un dominio personalizado. Para obtener más información, consulte Configuración de un dominio del grupo de usuarios.
2. Configure la versión de marca del dominio. Al crear o modificar el dominio, selecciona Inicio de sesión gestionado o Interfaz de usuario alojada (clásica). Esta opción establece el estilo de las páginas de inicio de sesión y se aplica a todos los clientes de aplicaciones que utilizan el dominio.
3. Cree un cliente de aplicación. En la pestaña Clientes de aplicaciones, crea un cliente de aplicaciones y establece su URL de retorno (también denominada URL de devolución de llamada o URI de redireccionamiento) en la URL desde la que se ejecuta la aplicación.

4. (Solo para el inicio de sesión gestionado) Personaliza el estilo. Abre el editor de marca del cliente de la aplicación para diseñar tus páginas de inicio de sesión. El editor de marca está disponible en un subconjunto de planes de funciones; para obtener más información, consulta. Planes de características de grupo de usuarios
5. Consulta tu página de inicio de sesión. En la pestaña Páginas de inicio de sesión de tu cliente de aplicación, selecciona Ver páginas de inicio de sesión para abrir las páginas de inicio de sesión gestionadas.

Configurar el inicio de sesión gestionado con AWS Amplify

Si lo utilizas AWS Amplify para añadir autenticación a tu aplicación web o móvil, puedes configurar tus páginas de inicio de sesión gestionadas en la interfaz de línea de comandos (CLI) de Amplify y las bibliotecas en el marco Amplify. Para añadir la autenticación a la aplicación, añada la categoría `Auth` al proyecto. A continuación, en su aplicación, autentique a los usuarios del grupo de usuarios con las bibliotecas cliente de Amplify.

Puede invocar páginas de inicio de sesión administrado para la autenticación o puede federar a los usuarios a través de un punto de conexión de autorización que redirija a un IdP. Después de que un usuario se autentique correctamente con el proveedor, Amplify crea un nuevo usuario en su grupo de usuarios y pasa los tokens del usuario a su aplicación.

Los siguientes ejemplos muestran cómo configurar el inicio AWS Amplify de sesión administrado con los proveedores sociales de tu aplicación.

- React
- Swift
- Flutter
- Android

Consulta de la página de inicio de sesión

En la consola de Amazon Cognito, pulse el botón Ver páginas de inicio de sesión en la pestaña Páginas de inicio de sesión de su cliente de aplicación, en el menú Clientes de aplicación. Este botón le llevará a una página de inicio de sesión en su dominio de grupos de usuarios 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 Ver página de inicio de sesión es útil cuando se quieren probar las funciones básicas de las páginas del inicio de sesión administrado. Sus páginas de inicio de sesión coinciden con la versión de marca que asignó al dominio de su grupo de usuarios. Puede personalizar la URL de inicio de sesión con parámetros adicionales y modificados. En la mayoría de los casos, los parámetros generados automáticamente en el enlace Ver página de inicio de sesión no se ajustan completamente a las necesidades de la 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 los puntos de conexión de grupos de usuarios y del inicio de sesión administrado.

La página web de inicio de sesión utiliza el siguiente formato URL. 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 relying-party url>
```

Puede buscar la cadena de dominio del grupo de usuarios desde el menú Dominio del grupo de usuarios. En el menú Clientes de aplicación, puede identificar los ID de los clientes de aplicación, sus URL de devolución de llamada, sus ámbitos permitidos y otras configuraciones.

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.

Ejemplo de solicitud de concesión implícita

Puede ver la página web de inicio de sesión 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://mydomain.auth.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://auth.example.com/
#id_token=eyJraaBcDeF1234567890&access_token=eyJraGhIjKlM1112131415&expires_in=3600&token_type=
```

Personalización de las páginas de autenticación

En el pasado, Amazon Cognito solo alojaba páginas de inicio de sesión con la interfaz de usuario alojada clásica, un diseño sencillo que otorga un aspecto universal a las páginas web de autenticación. Puede personalizar los grupos de usuarios de Amazon Cognito con una imagen de logotipo y modificar algunos estilos con un archivo que especifique algunos valores de estilo CSS. Más adelante, Amazon Cognito introdujo el inicio de sesión administrado, un nuevo servicio de autenticación alojado. El inicio de sesión administrado aporta un aspecto renovado con el editor de marcas. El editor de marcas es un editor visual sin código y ofrece un conjunto de opciones más amplio que la experiencia de personalización de la interfaz de usuario alojada. El inicio de sesión administrado también introdujo imágenes de fondo personalizadas y un tema de modo oscuro.

Puede cambiar entre las experiencias de inicio de sesión administrado y de personalización de la interfaz de usuario alojada en los grupos de usuarios. Para obtener más información sobre la personalización de las páginas de inicio de sesión administrado, consulte Aplicación de la creación de marca en las páginas de inicio de sesión administrado.

Cosas que debe saber sobre el inicio de sesión administrado y la interfaz de usuario alojada

La cookie de sesión de interfaz de usuario alojada y de inicio de sesión administrado de una hora de duración

Cuando un usuario inicia sesión con sus páginas de inicio de sesión o con un proveedor externo, Amazon Cognito establece una cookie en su navegador. Con esta cookie, los usuarios pueden volver a iniciar sesión con el mismo método de autenticación durante una hora. Cuando inician sesión

con la cookie de su navegador, obtienen nuevos tokens que duran el tiempo especificado en la configuración del cliente de aplicación. Los cambios en los atributos de los usuarios o en los factores de autenticación no afectan a su capacidad para volver a iniciar sesión con la cookie del navegador.

La autenticación con la cookie de sesión no restablece la duración de la cookie a una hora adicional. Los usuarios deben volver a iniciar sesión si intentan acceder a sus páginas de inicio de sesión más de una hora después de su última autenticación interactiva correcta.

Confirmación de las cuentas de usuario y verificación de los atributos de los usuarios

Para los usuarios locales del grupo de usuarios, el inicio de sesión administrado y la interfaz de usuario alojada funcionan mejor cuando se configura el grupo de usuarios en 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. En cambio, si confirma a los usuarios como administrador del grupo de usuarios, sus páginas de inicio de sesión mostrarán 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.

Ámbito de operaciones del inicio de sesión administrado

El inicio de sesión administrado y la interfaz de usuario alojada clásica admiten el registro, el inicio de sesión y la administración de contraseñas. Esto incluye completar el inicio de sesión con autenticación multifactor (MFA) y el registro de los autenticadores webAuthn. El inicio de sesión administrado no admite la administración de perfiles de autoservicio de los usuarios, como los cambios de atributos y la configuración de las preferencias de MFA. Debe implementar la administración de perfiles en el código de su propia aplicación. El inicio de sesión gestionado tampoco permite confirmar los cambios en los atributos al actualizar las direcciones de correo electrónico y los números de teléfono como administrador con la operación de la AdminUpdateUserAttributesAPI.

Consulta de los cambios realizados en la configuración

Si realiza cambios de estilo en sus páginas y estos no aparecen inmediatamente, espere unos minutos y 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. Puedes decodificar y verificar los tokens del grupo de usuarios mediante AWS Lambda. Consulte Decodificar y verificar los tokens JWT de Amazon Cognito en. GitHub

Versión de TLS

Las páginas de inicio de sesión administrado y de interfaz de usuario alojadas requieren el cifrado durante el tránsito. Los dominios de grupos de usuarios que Amazon Cognito proporciona requieren que los navegadores de los usuarios usen, como mínimo, la versión 1.2 de TLS. Los dominios personalizados admiten conexiones de navegador con la versión 1.2 de TLS. La interfaz de usuario alojada (clásica) no requiere TLS 1.2 para los dominios personalizados, pero el inicio de sesión administrado más reciente requiere la versión 1.2 de TLS, tanto para los dominios personalizados como para los de prefijo. Dado que Amazon Cognito gestiona la configuración de los servicios de dominio, no es posible modificar los requisitos TLS del dominio del grupo de usuarios.

Políticas CORS

Ni el inicio de sesión administrado ni la interfaz de usuario alojada admiten políticas de origen personalizadas para el uso compartido de recursos entre orígenes (CORS). Una política CORS impediría que los usuarios pasaran parámetros de autenticación en sus solicitudes. En su lugar, implemente una política CORS en el frontend de su aplicación. Amazon Cognito devuelve un encabezado de respuesta `Access-Control-Allow-Origin: *` a las solicitudes a los siguientes puntos de conexión.

1. Punto de conexión de token
2. Revocación de puntos de conexión
3. El punto de conexión userInfo

Cookies

El inicio de sesión administrado y la interfaz de usuario alojada configuran cookies en los navegadores de los usuarios. Las cookies cumplen los requisitos de algunos navegadores de que los sitios no instalen cookies de terceros. Están dirigidas únicamente a los puntos de conexión del grupo de usuarios e incluyen lo siguiente:

- Una cookie `XSRF-TOKEN` para cada solicitud.
- Una cookie `csrf-state` para garantizar la coherencia de la sesión cuando se redirige a un usuario.

- Una cookie `csrf-state-legacy` para garantizar la coherencia de la sesión, que Amazon Cognito lee como alternativa cuando el navegador no admite el atributo `SameSite`.
- Una cookie de sesión `cognito` que conserva los intentos de inicio de sesión correctos durante una hora.
- Una cookie `lang` que guarda la elección del usuario de idioma de localización en el inicio de sesión administrado.
- Una cookie `page-data` que guarda los datos necesarios mientras el usuario navega entre las páginas del inicio de sesión administrado.

En iOS, puede bloquear todas las cookies. Esta configuración no es compatible con la interfaz de usuario alojada ni el inicio de sesión administrado. Para trabajar con los usuarios que podrían habilitar esta configuración, cree la autenticación del grupo de usuarios en una aplicación iOS nativa con un AWS SDK. En este escenario, puede crear un almacenamiento de sesiones propio que no esté basado en cookies.

Efectos del cambio de versión del inicio de sesión administrado

Tenga en cuenta los siguientes efectos por añadir dominios y configurar la versión de inicio de sesión administrado.

- Cuando añade un dominio con prefijo, ya sea con inicio de sesión administrado o con una marca de interfaz de usuario alojada (clásica), las páginas de inicio de sesión pueden tardar hasta 60 segundos en estar disponibles.
- Cuando añade un dominio personalizado, ya sea con inicio de sesión administrado o con una marca de interfaz de usuario alojada (clásica), las páginas de inicio de sesión pueden tardar hasta 5 minutos en estar disponibles.
- Si cambia la versión de marca de su dominio, las páginas de inicio de sesión pueden tardar hasta 4 minutos en estar disponibles en la nueva versión de marca.
- Al cambiar entre el inicio de sesión administrado y la marca de la interfaz de usuario alojada (clásica), Amazon Cognito no mantiene las sesiones de usuario. El usuario debe volver a iniciar sesión con la nueva interfaz.

Estilo predeterminado

Al crear un cliente de aplicación en el Consola de administración de AWS, Amazon Cognito asigna automáticamente un estilo de marca al cliente de la aplicación. Al crear mediante programación un

cliente de aplicaciones con la CreateUserPoolClientoperación, no se crea ningún estilo de marca. El inicio de sesión administrado no está disponible para un cliente de aplicaciones creado con un AWS SDK hasta que crees uno con una CreateManagedLoginBrandingsolicitud.

Finalización del tiempo de espera de la petición de autenticación en el inicio de sesión administrado

Amazon Cognito cancela las solicitudes de autenticación que no se completan en 5 minutos y redirige al usuario al inicio de sesión administrado. La página muestra un mensaje de error `Something went wrong`.

Configuración de un dominio del grupo de usuarios

La configuración de un dominio es una parte opcional de la configuración de un grupo de usuarios. Un dominio de grupo de usuarios aloja características de autenticación de usuarios, federación con proveedores externos y flujos de OpenID Connect (OIDC). Cuenta con un inicio de sesión administrado, una interfaz precompilada para operaciones clave como el registro, el inicio de sesión o la recuperación de contraseñas. También aloja los puntos de conexión estándar de OpenID Connect (OIDC), como authorize, userInfo y token, para la autorización de máquina a máquina (M2M) y otros flujos de autenticación y autorización de OIDC y OAuth 2.0.

La autorización M2M también está disponible sin dominio. Puede solicitar los tokens de acceso a las credenciales del cliente con la operación de GetClientTokenAPI, que utiliza la API de Amazon Cognito en lugar del punto final del token.

Los usuarios se autentican con páginas de inicio de sesión administrado en el dominio asociado a su grupo de usuarios. Tienes dos opciones para configurar este dominio: puedes usar el dominio de prefijo de Amazon Cognito predeterminado o puedes configurar un dominio personalizado de tu propiedad.

La opción de dominio personalizado tiene más opciones de flexibilidad, seguridad y control. Por ejemplo, un dominio conocido y propiedad de una organización puede fomentar la confianza de los usuarios y hacer que el proceso de inicio de sesión sea más intuitivo. Sin embargo, el enfoque de dominio personalizado requiere algunos gastos adicionales, como la administración del certificado SSL o la configuración del DNS.

Los puntos de conexión de detección de OIDC, `/.well-known/openid-configuration` para las direcciones URL de los puntos de conexión y `/.well-known/jwks.json` para las claves de firma de token, no están alojados en su dominio. Para obtener más información, consulte Puntos de conexión de los proveedores de identidades y de la relación de confianza.

Comprender cómo configurar y administrar el dominio para su grupo de usuarios es un paso importante para integrar la autenticación en su aplicación. Sign-in utilizar la API de grupos de usuarios y un AWS SDK puede ser una alternativa a la configuración de un dominio. El API-based modelo entrega los tokens directamente en una respuesta de API, pero para las implementaciones que utilizan las capacidades ampliadas de los grupos de usuarios como un IdP de OIDC, debe configurar un dominio. Para obtener más información sobre los modelos de autenticación que están disponibles en los grupos de usuarios, consulte Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

Temas

- Cosas que debe saber acerca de los dominios de grupos de usuarios
- Uso del dominio de prefijo de Amazon Cognito para el inicio de sesión administrado
- Uso de un dominio propio con el inicio de sesión administrado

Cosas que debe saber acerca de los dominios de grupos de usuarios

Los dominios de grupos de usuarios son un punto de servicio para las relaciones de confianza de OIDC en las aplicaciones y para los elementos de la interfaz de usuario. Tenga en cuenta los siguientes detalles cuando planifique la implementación de un dominio para su grupo de usuarios.

Términos reservados

No puede usar el texto `aws`, `amazon` ni `cognito` en el nombre de un dominio con prefijo de Amazon Cognito.

Los puntos de conexión de detección se encuentran en otro dominio

Los puntos de conexión de detección `.well-known/openid-configuration` y `.well-known/jwks.json` no se encuentran en el dominio personalizado o con prefijo del grupo de usuarios. La ruta a estos puntos de conexión es la siguiente.

- `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/.well-known/openid-configuration`
- `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/.well-known/jwks.json`

Hora de entrada en vigor de los cambios de dominio

Amazon Cognito puede tardar hasta un minuto en lanzar o actualizar la versión de marca de un dominio de prefijo. Los cambios realizados en un dominio personalizado pueden tardar hasta cinco minutos en propagarse. Los nuevos dominios personalizados pueden tardar hasta una hora en propagarse.

Dominios personalizados y con prefijo al mismo tiempo

Puedes configurar un grupo de usuarios con un dominio personalizado y un dominio con prefijo que sea propiedad de. AWS Como los puntos de conexión de detección del grupo de usuarios están alojados en otro dominio, solo sirven al dominio personalizado. Por ejemplo, `openid-configuration` proporciona un valor único para`"authorization_endpoint"`. `"https://auth.example.com/oauth2/authorize"`

Si tiene dominios personalizados y con prefijo en un grupo de usuarios, puede usar el dominio personalizado con todas las características de un proveedor OIDC. El dominio de prefijo de un grupo de usuarios con esta configuración no tiene puntos de conexión clave de firma de tokens o de detección y debe usarse teniendo esto en cuenta.

Es preferible usar dominios personalizados como identificadores de actor de confianza en claves de acceso

Al configurar la autenticación del grupo de usuarios con claves de acceso, debe establecer un ID de actor de confianza (RP). Si tiene un dominio personalizado y un dominio de prefijo, puede configurar el ID de RP únicamente como su dominio personalizado. Para configurar un dominio de prefijo como ID de RP en la consola de Amazon Cognito, elimine su dominio personalizado o introduzca el nombre de dominio completo (FQDN) del dominio de prefijo como dominio. Third-party

No utilice dominios personalizados en distintos niveles de la jerarquía de dominios

Puede configurar grupos de usuarios distintos para tener dominios personalizados en el mismo dominio de nivel superior (TLD), por ejemplo, auth.example.com y auth2.example.com. La cookie de sesión del inicio de sesión administrado es válida para un dominio personalizado y todos los subdominios, por ejemplo, *.auth.example.com. Por este motivo, ningún usuario de las aplicaciones debe acceder al inicio de sesión administrado de ningún dominio ni subdominio principal. Cuando los dominios personalizados usen el mismo TLD, manténgalos en el mismo nivel de subdominio.

Supongamos que tiene un grupo de usuarios con el dominio personalizado auth.example.com. Luego crea otro grupo de usuarios y le asigna el dominio personalizado uk.auth.example.com. Un usuario inicia sesión en auth.example.com y obtiene una cookie que su navegador presenta a

cualquier sitio web en la ruta comodín *.auth.example.com. A continuación, intenta iniciar sesión en uk.auth.example.com. Envían una cookie no válida a su dominio de grupo de usuarios y reciben un error en lugar de una solicitud de inicio de sesión. Por el contrario, un usuario con una cookie para *.auth.example.com no tendrá problemas para iniciar sesión en auth2.example.com.

Versión de marca

Al crear un dominio, está configurando una versión de marca. Sus opciones son la nueva experiencia de inicio de sesión administrado y la clásica experiencia de interfaz de usuario alojada. Esta opción se aplica a todos los clientes de aplicaciones que alojan servicios en su dominio.

Uso del dominio de prefijo de Amazon Cognito para el inicio de sesión administrado

La experiencia predeterminada para el inicio de sesión gestionado se aloja en un dominio propietario. AWS Este enfoque tiene pocas barreras de entrada (basta con elegir un nombre de prefijo y estará activo), pero no cuenta con las características que inspiran confianza de un dominio personalizado. El costo es el mismo para ambas opciones de dominio. Solo se diferencian por el dominio de la dirección web a la que dirige a los usuarios. En los casos de redireccionamientos de IdP de terceros y flujos de credenciales de clientes, el dominio del prefijo tiene poco efecto visible. Un dominio personalizado es mejor para los casos en los que los usuarios inician sesión con un inicio de sesión administrado e interactúan con un dominio de autenticación que no coincide con el dominio de la aplicación.

El dominio de prefijo tiene el prefijo que elija y es un subdominio del dominio de autenticación regional`amazoncognito.com`. A continuación, se muestra un ejemplo:

```
https://cognitoexample.auth.ap-south-1.amazoncognito.com
```

Todos los dominios de prefijo tienen el formato *`prefix`*`.auth.`*`Región de AWS code`*`.amazoncognito.com`. Los grupos de usuarios de dominios personalizados pueden alojar la interfaz de usuario alojada o el inicio de sesión administrado en cualquier dominio que sea de su propiedad.

> **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 los grupo de usuarios adoptan los siguientes formatos.

```
auth.Region.amazoncognito.com
auth-fips.Region.amazoncognito.com
```

El `auth-fips` dominio es un punto final que cumple con la norma federal de procesamiento de información (FIPS) 140-3. Proporciona cifrado en tránsito con módulos FIPS-validated criptográficos. Dirija a sus usuarios a este punto final cuando su aplicación deba cumplir los requisitos de conformidad con la FIPS. Los puntos de enlace FIPS para el inicio de sesión gestionado de Amazon Cognito están disponibles en un subconjunto AWS de regiones de EE. UU. y no se ofrecen en todas las regiones. Para obtener más información sobre FIPS y las regiones en las que Amazon Cognito proporciona puntos de enlace de FIPS, consulte FIPS 140-3.

Para añadir un cliente de aplicaciones y un dominio de grupo de usuarios con el, consulte. Consola de administración de AWSCreación de un cliente de aplicación

Temas

- Requisitos previos
- Configuración de un prefijo de dominio de Amazon Cognito
- Verificación de 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 Creación de una nueva aplicación en la consola de Amazon Cognito.

Configuración de un prefijo de dominio de Amazon Cognito

Puede usar la API Consola de administración de AWS o la AWS CLI o para configurar un dominio de grupo de usuarios.

Amazon Cognito console

Configuración de un dominio

1. Vaya al menú Dominio en Creación de marca.
2. Junto a Dominio, elija Acciones y seleccione Crear dominio de Cognito. Si ya ha configurado un dominio de prefijo del grupo de usuarios, seleccione Eliminar dominio de Cognito antes de crear su nuevo dominio personalizado.
3. Ingrese un prefijo de dominio disponible para utilizarlo con un dominio de Amazon Cognito. Para obtener más información sobre cómo configurar un dominio personalizado, consulte Uso de un dominio propio con el inicio de sesión administrado.
4. Elija una Versión de marca. Su versión de marca se aplica a todas las páginas interactivas para el usuario de ese dominio. Su grupo de usuarios puede alojar el inicio de sesión administrado o la marca de interfaz de usuario alojada para todos los clientes de aplicación.

   > **Note**
   >
   > Puede tener un dominio personalizado y un dominio de prefijo, pero Amazon Cognito solo servirá el punto de conexión `/.well-known/openid-configuration` para el dominio personalizado.

5. 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> --managed-login-version 2`
- Funcionamiento de la API de grupos de usuarios: 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>`

- Funcionamiento de la API de grupos de usuarios: DescribeUserPoolDomain

Eliminación de un dominio

- AWS CLI: `aws cognito-idp delete-user-pool-domain`

  Ejemplo: `aws cognito-idp delete-user-pool-domain --domain <domain_name>`
- Funcionamiento de la API de grupos de usuarios: DeleteUserPoolDomain

Verificación de 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 el inicio de sesión administrado

Después de configurar un cliente de aplicación, puede configurar su grupo de usuarios con un dominio personalizado para los servicios de dominio del inicio de sesión administrado. Con un dominio personalizado, los usuarios pueden iniciar sesión en tu aplicación con tu propia dirección web en lugar del dominio de `amazoncognito.com` prefijo predeterminado. Los dominios personalizados mejoran la confianza de los usuarios en una aplicación con un nombre de dominio conocido, especialmente cuando el dominio raíz coincide con el dominio que aloja la aplicación. Los dominios personalizados pueden mejorar el cumplimiento de los requisitos de seguridad de la organización.

Un dominio personalizado tiene algunos requisitos previos, como un grupo de usuarios, un cliente de aplicación y un dominio web de su propiedad. Los dominios personalizados también requieren un

certificado SSL para el dominio personalizado, gestionado con AWS Gestor de certificados (ACM) en EE. UU. Este (Norte de Virginia). Amazon Cognito crea una CloudFront distribución de Amazon, asegurada en tránsito con su certificado ACM. Como usted es el propietario del dominio, debe crear un registro DNS que dirija el tráfico a la CloudFront distribución de su dominio personalizado.

Cuando estos elementos estén listos, puede añadir el dominio personalizado a su grupo de usuarios a través de la API o la consola de Amazon Cognito. Esto implica especificar el nombre de dominio y el certificado SSL y, a continuación, actualizar la configuración de DNS con el alias de destino proporcionado. Después de realizar estos cambios, puede comprobar si se puede acceder a la página de inicio de sesión desde el dominio personalizado.

La forma más sencilla de crear un dominio personalizado es con una zona alojada pública en Amazon Route 53. La consola de Amazon Cognito puede crear los registros DNS correctos en unos pocos pasos. Antes de empezar, valore la posibilidad de crear una zona alojada de Route 53 para un dominio o subdominio de su propiedad.

Temas

- Adición de un dominio personalizado a un grupo de usuarios
- Requisitos previos
- Paso 1: Introducir el nombre de dominio personalizado
- Paso 2: Agregar un destino de alias y un subdominio
- Paso 3: Verificar la página de inicio de sesión
- Cambio del certificado SSL en el dominio personalizado
- Establecer la versión mínima de TLS para su 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 Gestor de certificados (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 Creación de una nueva aplicación en la consola de Amazon Cognito.

- 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. Le recomendamos auth para el nombre de su 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 SSL/TLS certificado público gestionado por ACM en el este de EE. UU. (Virginia del Norte). El certificado debe estar en us-east-1 porque está asociado a una distribución CloudFront en un servicio global.
- Clientes de navegador compatibles con la indicación del nombre de servidor (SNI). La CloudFront distribución que Amazon Cognito asigna a los dominios personalizados requiere el SNI. No se puede cambiar esta configuración. Para obtener más información sobre el SNI en CloudFront las distribuciones, consulte Usar el SNI para atender solicitudes HTTPS en la Guía para desarrolladores de Amazon CloudFront .
- Una aplicación que permite al servidor de autorización del grupo de usuarios añadir cookies a las sesiones de los usuarios. Amazon Cognito establece varias cookies obligatorias para las páginas del inicio de sesión administrado. 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 del inicio de sesión administrado aumenten de tamaño. Un servicio intermedio, como un equilibrador de carga de aplicación (ALB), delante del dominio personalizado puede imponer un tamaño máximo de encabezado o un tamaño total de cookies. Si la aplicación también establece sus propias cookies, es posible que las sesiones de los usuarios superen estos límites. Le recomendamos que, para evitar conflictos con los límites de tamaño, su aplicación no establezca cookies en el subdominio que aloja los servicios de dominio de su grupo de usuarios.
- 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:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Sid": "AllowCloudFrontUpdateDistribution",
            "Effect": "Allow",
            "Action": [
                "cloudfront:updateDistribution"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Para obtener más información sobre la autorización de acciones en CloudFront, consulte Uso de Identity-Based políticas (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. Puede seleccionar la versión mínima de TLS para la política de seguridad de distribución al crear o actualizar su dominio personalizado con la API de Amazon Cognito. Para obtener más información, consulte Establecer la versión mínima de TLS para su dominio personalizado.

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. Vaya al menú Dominio en Creación de marca.
2. Junto a Dominio, elija Acciones y seleccione Crear dominio personalizado o Crear dominio de Amazon Cognito. Si ya ha configurado un dominio personalizado del grupo de usuarios, seleccione Eliminar dominio personalizado antes de crear su nuevo dominio personalizado.

3. Junto a Dominio, elija Acciones y seleccione Crear dominio personalizado. Si ya ha configurado un dominio personalizado, elija Eliminar dominio personalizado para eliminar el dominio existente antes de crear el nuevo dominio personalizado.
4. 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.
5. 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 Gestor de certificados .
6. Elija una Versión de marca. Su versión de marca se aplica a todas las páginas interactivas para el usuario de ese dominio. Su grupo de usuarios puede alojar el inicio de sesión administrado o la marca de interfaz de usuario alojada para todos los clientes de aplicación.

   > **Note**
   >
   > Puede tener un dominio personalizado y un dominio de prefijo, pero Amazon Cognito solo servirá el punto de conexión `/.well-known/openid-configuration` para el dominio personalizado.
7. Seleccione Crear.
8. Amazon Cognito lo devuelve al menú Dominio. 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

El siguiente cuerpo de CreateUserPoolDomainsolicitud crea un dominio personalizado.

```
{
   "Domain": "auth.example.com",
```

```
   "UserPoolId": "us-east-1_EXAMPLE",
   "ManagedLoginVersion": 2,
   "CustomDomainConfig": {
    "CertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "SecurityPolicy": "TLS_V1_3_2025"
   }
}
```

El `SecurityPolicy` campo opcional establece la versión mínima de TLS para tu dominio personalizado. Para obtener más información, consulte Establecer la versión mínima de TLS para su dominio personalizado.

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, introduzca sus AWS credenciales.
2. Si no dispone de una zona alojada pública en Route 53, cree una con una raíz que sea la principal de su dominio personalizado. Para obtener más información, consulte Creating a public hosted zone en la Guía para desarrolladores de Amazon Route 53.
   a. Elija Create Hosted Zone (Crear zona alojada).
   b. Introduzca el dominio principal, por ejemplo*auth.example.com*, de su dominio personalizado, por ejemplo*myapp.auth.example.com*, de la lista de nombres 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 con una delegación establecida 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 en 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. Agregue un registro DNS para el dominio principal de su dominio personalizado, si aún no dispone de uno. Cree un registro de DNS para el dominio principal con las siguientes propiedades:

   - Nombre del registro: déjelo en blanco.
   - Tipo de registro: `A`.
   - Alias: no lo habilite.
   - Valor: introduzca un objetivo de su elección. Este registro debe convertirse en algo, pero el valor del registro no le importa a Amazon Cognito.
   - TTL: configúrelo en el TTL que prefiera o déjelo como predeterminado.
   - Política de direccionamiento: elija Direccionamiento sencillo.

5. Elija Crear registros. El siguiente es un ejemplo de registro para el dominio*example.com*:

   ```
   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 de DNS para el dominio principal, Amazon Cognito

devuelve un error cuando intenta configurar el dominio personalizado. Un registro de inicio de autoridad (SOA) no es un registro de DNS suficiente para la verificación del dominio principal.

6. Agregue otro registro de DNS para el dominio personalizado con las siguientes propiedades:

   - Nombre de registro: el prefijo de dominio personalizado; por ejemplo, `auth` para crear un registro para `auth.example.com`.
   - Tipo de registro: `A`.
   - Alias: habilítelo.
   - Dirigir el tráfico a: elija Alias de la distribución de CloudFront. Introduzca el Destino de alias registrado anteriormente, por ejemplo, `123example.cloudfront.net`.
   - Política de direccionamiento: elija Direccionamiento sencillo.

7. Elija 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 *`auth`* subdominio:

```
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 Gestor de certificados .
> 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.

Consola de administración de AWS

Para renovar un certificado mediante la consola de Amazon Cognito:

1. Inicie sesión en la consola de Amazon Cognito Consola de administración de AWS 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 el menú Dominio.
5. Elija Actions (Acciones), Edit ACM certificate (Editar certificado de ACM).
6. Seleccione el nuevo certificado que desea asociar a su dominio personalizado.

7. Seleccione Save changes (Guardar cambios).

API

Para renovar un certificado (API de Amazon Cognito)

- Utilice la acción UpdateUserPoolDomain.

Establecer la versión mínima de TLS para su dominio personalizado

Al crear o actualizar un dominio personalizado con la consola o la API de Amazon Cognito, puede seleccionar la versión mínima de TLS para la distribución de Amazon. CloudFront La distribución usa esta versión cuando se comunica con los clientes. En la API, establece la versión mínima de TLS en el `SecurityPolicy` campo del `CustomDomainConfig` objeto de una UpdateUserPoolDomainsolicitud CreateUserPoolDomainor. La política de seguridad define la versión mínima de TLS y los conjuntos de cifrado que se CloudFront admiten en la comunicación con los clientes. Para obtener orientación específica, consulte Protocolos y cifrados compatibles entre espectadores y CloudFront en la Guía para CloudFront desarrolladores de Amazon. Puede seleccionar uno de los siguientes valores.

`TLS_V1_3_2025`(el más estricto)

Una política preparada para la poscuántica que requiere TLS 1.3. Ofrece el mejor nivel de seguridad y es ideal para cargas de trabajo en las que todos los clientes y navegadores están actualizados a las versiones más recientes. Para obtener más información, consulte Protocolos y cifrados compatibles con _2025. TLSv1.3

`TLS_V1_2_2021` (recomendado)

Una política preparada para la poscuántica que prefiere el TLS 1.3, pero permite recurrir al TLS 1.2 para adaptarse a los clientes más antiguos. Es el mínimo recomendado para las aplicaciones de consumo típicas de calidad comercial. Para obtener más información, consulte Protocolos y cifrados compatibles para _2021. TLSv1.2

`TLS_V1`(se desaconseja encarecidamente)

Permite recurrir a TLS 1.0. Ofrece la compatibilidad más amplia, incluida la compatibilidad con clientes antiguos que tienen más de una década de antigüedad. Esta compatibilidad se consigue a costa de permitir el uso comercial de versiones de TLS y algoritmos criptográficos que ya no se

consideran seguros. Para obtener más información, consulte Protocolos y cifrados compatibles con TLSv1.

> **Note**
>
> Los dominios personalizados que utilizan el inicio de sesión administrado (versión de marca 2) requieren una versión TLS mínima igual o superior. `TLS_V1_2_2021` Una solicitud que establece un dominio `TLS_V1` de inicio de sesión administrado devuelve un `InvalidInputException` error.

Puede seleccionar la versión mínima de TLS desde la consola de Amazon Cognito o con la API de Amazon Cognito, AWS Command Line Interface el AWS CLI() y los SDK. AWS

El siguiente cuerpo de la UpdateUserPoolDomainsolicitud actualiza la versión mínima de TLS de un dominio personalizado existente.

```
{
   "Domain": "auth.example.com",
   "UserPoolId": "us-east-1_EXAMPLE",
   "ManagedLoginVersion": 2,
   "CustomDomainConfig": {
    "CertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "SecurityPolicy": "TLS_V1_3_2025"
   }
}
```

Aplicación de la creación de marca en las páginas de inicio de sesión administrado

Es posible que desee ofrecer una experiencia de usuario uniforme entre el servicio de autenticación y la aplicación. Puede lograr este objetivo con formularios personalizados y operaciones de API de back-end en un AWS SDK, o con un inicio de sesión administrado. El inicio de sesión administrado y la clásica interfaz de usuario alojada son interfaces web para el componente de la aplicación que sirve de autenticación con grupos de usuarios. Para sincronizar los servicios de autenticación administrada con la experiencia de usuario de la aplicación, dispone de dos opciones de personalización: el editor de marcas y la marca de interfaz de usuario alojada. Puede elegir la

experiencia que prefiera en la consola de Amazon Cognito y con las operaciones de API del grupo de usuarios.

El editor de marcas

El editor de marcas es la opción de personalización más reciente para la nueva experiencia de interfaz de usuario de grupos de usuarios, el inicio de sesión administrado. El editor de marcas es un editor visual sin código para los activos y el estilo de inicio de sesión administrados, y un conjunto de operaciones de API para la configuración programática de un gran número de opciones de configuración. Los grupos de usuarios que configura con un dominio y un inicio de sesión administrado muestran automáticamente la versión de diseño de marca de sus páginas de inicio de sesión.

Marca de interfaz de usuario alojada (clásica)

La experiencia de marca de interfaz de usuario alojada (clásica) tiene dos opciones: modificar un archivo de hojas de estilo en cascada (CSS) con un conjunto fijo de opciones de estilo y añadir una imagen de logotipo personalizada. Puede configurar estas opciones en la consola de Amazon Cognito o con la operación de la API SetUICustomization. Cuando se lanzó el servicio, Amazon Cognito solo tenía esta opción. Los grupos de usuarios que configure con un dominio y la versión de marca de interfaz de usuario alojada representan automáticamente la versión clásica de sus páginas de inicio de sesión. Es posible que su plan de características también admita solo la interfaz de usuario alojada.

> **Note**
>
> El editor de marcas y la experiencia de marca clásica modifican las propiedades visuales del servicio de autenticación alojado. Actualmente, no puede modificar el texto que se muestra en las páginas de inicio de sesión administrado, excepto para aplicarle la localización en uno de los distintos idiomas. Para obtener más información sobre configuraciones locales, consulte Localización de inicio de sesión administrado.

Elección de una experiencia de marca y asignación de estilos

En la consola de Amazon Cognito, los nuevos grupos de usuarios utilizan de forma predeterminada la experiencia de creación de marca Inicio de sesión administrado. Los grupos de usuarios que configuró antes de que estuviera disponible el inicio de sesión gestionado tienen la marca Hosted

UI (clásica). Puede cambiar entre el inicio de sesión administrado y la marca de interfaz de usuario alojada. Cuando cambia su versión de marca, Amazon Cognito aplica el cambio inmediatamente a las páginas interactivas para el usuario del dominio de su grupo de usuarios. Con el inicio de sesión administrado y la interfaz de usuario alojada, su grupo de usuarios puede tener un estilo para cada cliente de aplicación.

Cada cliente de aplicación puede tener un estilo de marca distinto, pero el dominio de un grupo de usuarios sirve tanto para el inicio de sesión administrado como para la interfaz de usuario alojada. Un estilo es el conjunto de ajustes de personalización que se aplica a un cliente de aplicación. Puede configurar un dominio personalizado y un dominio con prefijo por cada grupo de usuarios. Puede asignar diferentes versiones de marca a sus dominios personalizados y con prefijo. Sin embargo, un dominio con prefijo no es completamente funcional cuando también tiene un dominio personalizado: los puntos de conexión de detección del OIDC `.well-known` solo presentan rutas de dominio personalizadas. Solo puede usar el dominio con prefijo para operaciones que no requieran la detección de puntos de conexión (`openid-configuration`) en un grupo de usuarios con esta configuración. Gracias a estas propiedades de los grupos de usuarios, puede elegir de forma eficaz una versión de marca por grupo de usuarios.

Puede asignar estilos a los clientes de la aplicación en un grupo de usuarios en el que un dominio esté configurado para la versión de marca de inicio de sesión administrado. Los estilos son un conjunto de ajustes visuales compuesto por archivos de imagen, opciones de visualización y valores CSS. Al asignar un estilo a un cliente de aplicación, Amazon Cognito envía inmediatamente las actualizaciones a las páginas de inicio de sesión interactivas para el usuario. Amazon Cognito renderiza sus páginas interactivas para el usuario con la versión de marca que haya elegido y la personalización que le haya aplicado.

Actualización y eliminación de estilos

Al crear un estilo, lo vincula a un cliente de aplicación. Para cambiar una asignación de estilo para un cliente de aplicación, primero debe eliminar el estilo original. En la actualidad, no se pueden copiar ajustes entre estilos. Debe hacerlo mediante programación. Para replicar la configuración entre estilos y clientes de aplicaciones, obtenga la configuración de un estilo con la operación de la DescribeManagedLoginBrandingAPI y aplíquela con CreateManagedLoginBrandingo UpdateManagedLoginBranding. No puede cambiar los estilos asignados a un cliente de aplicación; solo puede eliminar el original y establecer uno nuevo. Para obtener más información sobre cómo gestionar estilos con operaciones API y SDK, consulte Operaciones de API y SDK para la creación de marcas de inicio de sesión administrado.

> **Note**
>
> Las solicitudes programáticas que crean o actualizan un estilo de marca no deben tener un tamaño superior a 2 MB. Si su solicitud supera este límite, divídala en varias solicitudes `UpdateManagedLoginBranding` para grupos de parámetros que no superen el tamaño máximo de la solicitud. Estas solicitudes no dan como resultado que los parámetros no especificados se establezcan de forma predeterminada, por lo que puede enviar solicitudes parciales sin que ello afecte a la configuración existente.

Para eliminar un estilo, debe ir a la consola de Amazon Cognito desde el menú Inicio de sesión administrado. En Estilos, elija el estilo que desee eliminar y Eliminar estilo.

En general, el proceso de asignación de la marca a un dominio consta de los pasos siguientes.

1. Crear un dominio y configurar la versión de marca.
2. Crear un estilo de marca y asignarlo a un cliente de aplicación.

Cómo asignar un estilo a un cliente de aplicación

1. En el menú Dominio de su grupo de usuarios, cree un dominio y configure la Versión de marca como Inicio de sesión administrado.
2. Navegue hasta el menú Inicio de sesión administrado. En Estilos, seleccione Crear un estilo.
3. Elija el cliente de aplicación al que quiere asignar su estilo o cree un nuevo cliente de aplicación.
4. Para empezar a configurar los ajustes de marca, seleccione Iniciar el editor de marca.

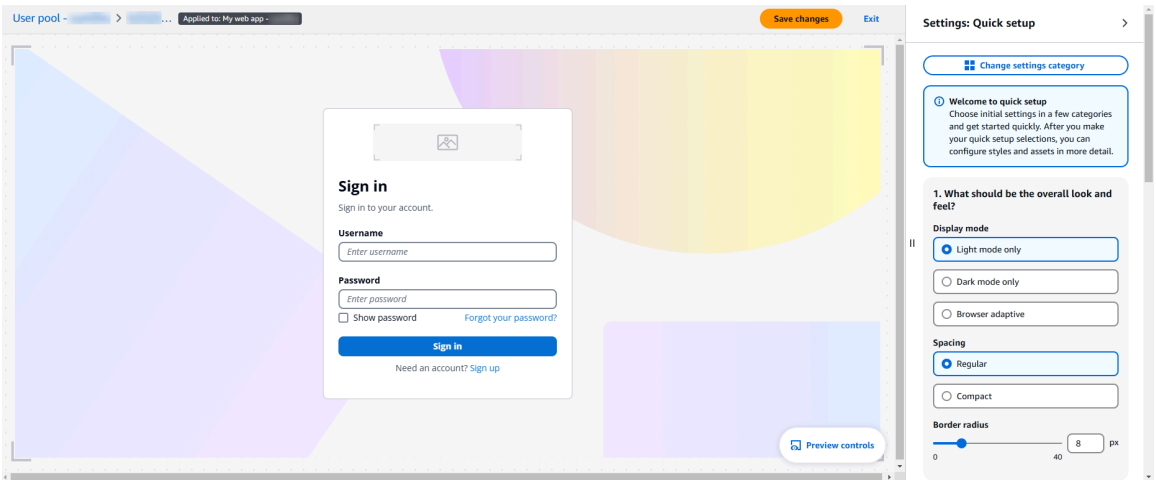
Temas

- El editor de marca y la personalización del inicio de sesión administrado

El editor de marca y la personalización del inicio de sesión administrado

El editor de marcas es una herramienta visual de diseño y edición para sus páginas web de inicio de sesión administrado. Está integrado en la consola Amazon Cognito. En el editor de marcas, puede empezar con una vista previa de sus páginas de inicio de sesión y pasar a una opción de configuración rápida o a una vista detallada con opciones avanzadas. Puede modificar y

previsualizar los parámetros de estilo o añadir una imagen de fondo y un logotipo personalizados. Puede configurar el modo claro y el modo oscuro.



Para empezar, cree un estilo que pueda aplicar a su grupo de usuarios o a un cliente de aplicación.

Cómo empezar a usar el editor de marcas

1. Cree un dominio desde la pestaña Dominio o actualice su dominio actual. En Versión de marca, configure el dominio para que utilice el Inicio de sesión administrado.
2. Elimine el estilo de cliente de aplicación existente si lo hubiera.
   a. En el menú Clientes de aplicación, seleccione el cliente de aplicación.
   b. En Estilo de inicio de sesión administrado, seleccione el estilo asignado al cliente de su aplicación.
   c. Elija Eliminar estilo. Confirme la opción elegida.
3. Navegue hasta el menú Inicio de sesión administrado en su grupo de usuarios. Si aún no lo ha hecho, siga las instrucciones para seleccionar un plan de características que incluya el inicio de sesión administrado. También puede seleccionar Vista previa de esta característica si quiere echar un vistazo al editor de marcas sin hacer cambios.
4. En Estilos, seleccione Crear un estilo.
5. Elija el cliente de aplicación al que desea asignar su estilo y seleccione Crear. También puede crear un cliente de aplicación nuevo.
6. La consola de Amazon Cognito lanza el editor de marcas.
7. Elija la pestaña en la que desee empezar a editar o seleccione Iniciar el editor e introduzca la configuración rápida. Las siguientes pestañas están disponibles:

Vista previa

Compruebe cuáles son sus selecciones actuales en las páginas de inicio de sesión administrado.

Bases

Establezca un tema general, configure los enlaces a proveedores de identidad externos y aplique estilos a los campos de los formularios.

Componentes

Configure estilos para encabezados, pies de página y elementos individuales de la interfaz de usuario.

8. Para tomar decisiones sobre la configuración inicial, introduzca la configuración rápida. Seleccione Cambiar la categoría de configuración y elija Configuración rápida. Al seleccionar Proceder, se abre el editor de marcas con un conjunto de opciones básicas que puede configurar.

Texto y localización

No es posible modificar ni localizar el texto en el editor de marcas. En lugar de eso, añada un parámetro de consulta `lang` a la URL que distribuye a los usuarios. Este parámetro hace que las páginas de inicio de sesión administrado se traduzcan a uno de los varios idiomas disponibles. Para obtener más información, consulte Localización de inicio de sesión administrado.

Configuración rápida

El botón Iniciar el editor de marcas carga un editor visual para la configuración de inicio de sesión gestionado, en el que puede seleccionar entre una variedad de opciones de personalización principales. A medida que realiza selecciones, Amazon Cognito muestra los cambios de inicio de sesión administrados en una ventana de vista previa. Para volver al menú de configuración detallado, seleccione el botón Cambiar la categoría de configuración.

En términos generales, ¿cuál debería ser el aspecto y qué sensación debería transmitir?

Configure los ajustes básicos del tema para el inicio de sesión administrado.

Modo de visualización

Elija una experiencia de modo claro, modo oscuro o adaptativa para su inicio de sesión administrado. La configuración adaptativa depende de la preferencia del navegador del usuario cuando Amazon Cognito procesa el inicio de sesión administrado. Si elige un modo adaptable al navegador, puede elegir diferentes colores e imágenes de logotipos para el modo claro y oscuro.

Spacing

Establezca el espaciado predeterminado entre los elementos de la página.

Radio del borde

Defina la profundidad de redondeo del borde exterior de los elementos.

¿Cómo debe quedar el fondo de la página?

Tipo de fondo

La casilla Mostrar imagen indica si desea una imagen de fondo o establecer un color de fondo sólido.

1. Para usar una imagen, seleccione Mostrar imagen y elija una imagen de fondo para los modos claro y oscuro. También puede establecer un Color de fondo de la página en modo oscuro y modo claro para las áreas del fondo que no estén cubiertas por la imagen.
2. Para usar solo un color de fondo, quite la selección de Mostrar imagen y elija un color de fondo de la página en modo claro y oscuro.

¿Qué aspecto deben tener los formularios?

Configure los ajustes de los elementos del formulario del inicio de sesión administrado. Algunos ejemplos de elementos de formulario son las solicitudes de inicio de sesión y las solicitudes de código.

Alineación horizontal

Establezca la alineación horizontal de los campos del formulario.

Logotipo del formulario

Establezca el posicionamiento de la imagen de su logotipo.

Imagen del logotipo

Elija un archivo de imagen del logotipo para incluirlo en el elemento del formulario para los modos claro y oscuro. Para cargar una imagen, seleccione el menú desplegable Imagen del logotipo, elija Agregar nuevo activo y añada un archivo de logotipo.

Color de marca principal

Establezca un color de tema para los modos claro y oscuro. Este color se aplica como color de fondo a todos los elementos clasificados como primarios.

¿Qué aspecto deben tener los encabezados?

Elija si quiere incluir un encabezado en sus páginas de inicio de sesión administrado. El encabezado puede contener una imagen de logotipo.

Logotipo del encabezado

Establezca la posición de la imagen del logotipo en su encabezado.

Imagen del logotipo

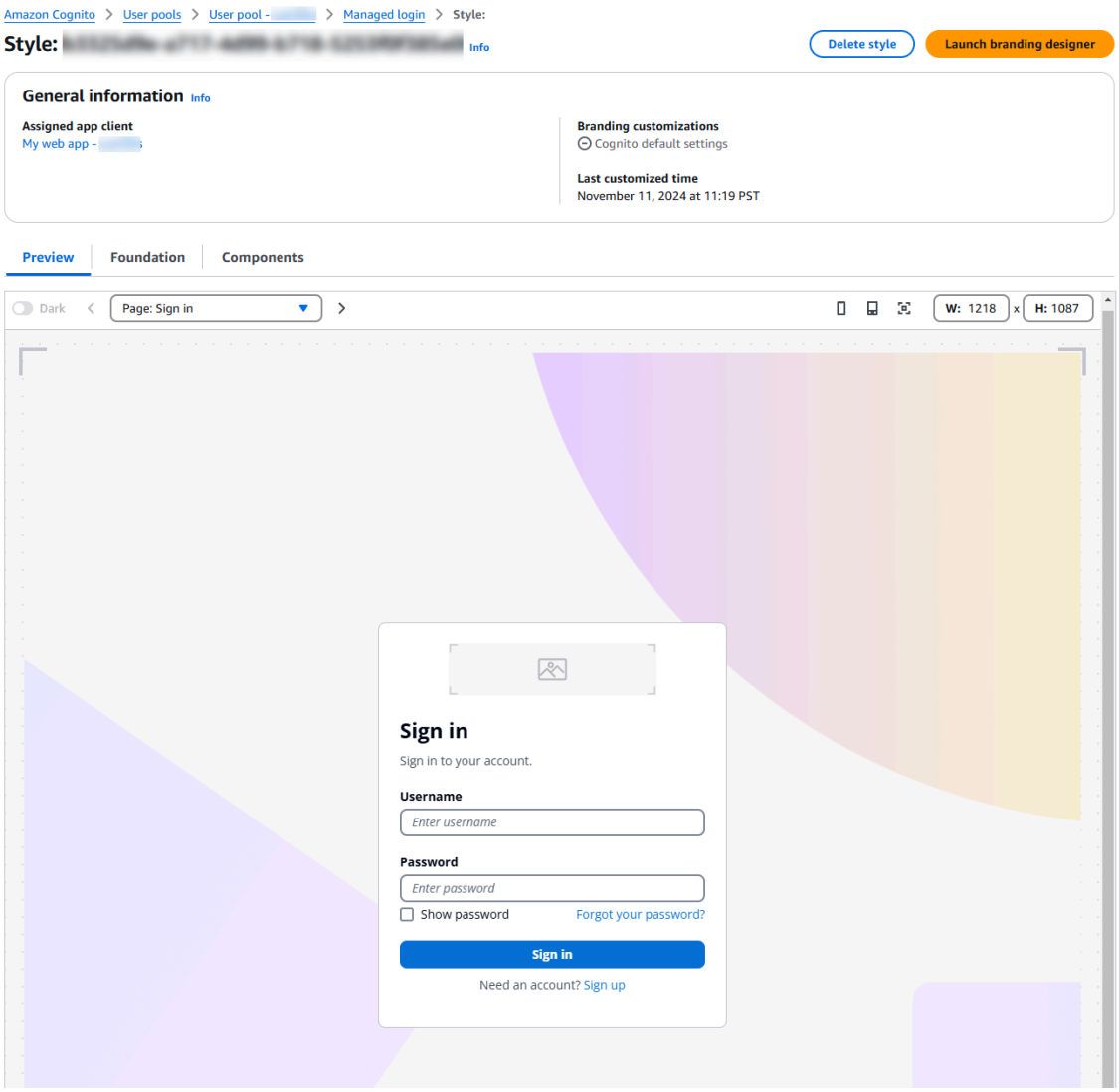
Elija una posición del logotipo y un archivo de imagen del logotipo para incluirlos en el encabezado. Para cargar una imagen, seleccione el menú desplegable Imagen del logotipo, seleccione Agregar nuevo activo y añada un archivo de logotipo.

Color de fondo del encabezado

Configure los colores de los modos claro y oscuro para el fondo del encabezado.

Configuración detallada

En la vista de configuración detallada, puede modificar los componentes individuales de la base y los componentes. La pestaña Vista previa muestra una vista previa del inicio de sesión administrado en el contexto actual con sus personalizaciones.



Para acceder al editor visual de un componente, elija el icono de edición en el mosaico del componente. Desde el editor del estudio de temas, puede cambiar de un componente a otro con el botón Cambiar la categoría de configuración.

Bases

Estilos de aplicación

Configure los aspectos básicos de su configuración de inicio de sesión administrado. Esta categoría tiene ajustes para el tema general, el espaciado del texto y el encabezado y el pie de página.

Modo de visualización

Elija una experiencia de modo claro, modo oscuro o adaptativa para sus páginas de inicio de sesión. Si elige un modo adaptable al navegador, puede elegir diferentes colores e imágenes de logotipos para el modo claro y oscuro.

Spacing

Establezca el espaciado predeterminado entre los elementos de la página.

Comportamiento de la autenticación

Configure los estilos de los botones que conectan a sus usuarios con proveedores de identidad externos (IdPs). Esta sección incluye la opción Entrada de búsqueda de dominio para que el administrador solicite a los usuarios una dirección de correo electrónico y la compare con su identificador de proveedor de identidad SAML.

Comportamiento del formulario

Configure los estilos de los formularios de entrada: la posición de las entradas, los colores y la alineación de los elementos.

Componentes

Botones

Estilos de los botones que Amazon Cognito representa en las páginas de inicio de sesión administrado.

Divisor

Estilos para las líneas divisorias y los límites entre los elementos de inicio de sesión administrado, como el formulario de entrada y el selector de inicio de sesión del proveedor externo.

Lista desplegable

Estilos para menús desplegables.

Icono de favoritos

Estilos para la imagen que Amazon Cognito proporciona para la pestaña y el icono del marcador.

Anillos de enfoque

Estilos para los aspectos más destacados que indican una entrada actualmente seleccionada.

Contenedor de formulario

Estilos de los elementos que delimitan un formulario.

Pie de página global

Estilos para el pie de página que Amazon Cognito muestra en la parte inferior de las páginas de inicio de sesión administrado.

Encabezado global

Estilos para el encabezado que Amazon Cognito muestra en la parte superior de las páginas de inicio de sesión administrado.

Indicaciones

Estilos para los mensajes de error y de éxito.

Controles de opciones

Estilos para casillas de verificación, selecciones múltiples y otras peticiones de entrada.

Fondo de página

Estilos para el contexto general del inicio de sesión administrado.

Entradas

Estilos para las solicitudes de entrada de campos de formulario.

Link

Estilos de hipervínculos en las páginas de inicio de sesión administrado.

Texto para página

Estilos para el texto de la página.

Texto para campo

Estilos del texto alrededor de las entradas del formulario.

Operaciones de API y SDK para la creación de marcas de inicio de sesión administrado

También puede aplicar la marca a un estilo de inicio de sesión administrado con las operaciones de la API CreateManagedLoginBrandingy UpdateManagedLoginBranding. Estas operaciones son ideales para crear versiones idénticas o ligeramente modificadas de un estilo de marca para otra aplicación, cliente o grupo de usuarios. Consulta la marca de inicio de sesión gestionado de un estilo existente con la operación de API y DescribeManagedLoginBranding, a continuación, modifica el resultado según sea necesario y aplícalo a otro recurso.

La operación `UpdateManagedLoginBranding` no cambia el cliente de aplicación al que se aplica su estilo. Solo actualiza el estilo existente que está asignado a un cliente de aplicación. Para reemplazar por completo el estilo de un cliente de aplicación, elimina el estilo existente con DeleteManagedLoginBrandingy asigna un estilo nuevo con`CreateManagedLoginBranding`. En la consola de Amazon Cognito, ocurre lo mismo: debe eliminar el estilo existente y crear uno nuevo.

Para configurar una marca de inicio de sesión administrado en una solicitud de API o SDK, es necesario que la configuración esté integrada en un archivo JSON que se convierta en un tipo de datos `Document`. A continuación, encontrará directrices sobre las imágenes que puede añadir y para generar solicitudes programáticas a fin de configurar un estilo de marca.

Administración de imágenes

CreateManagedLoginBrandinge UpdateManagedLoginBrandingincluye un `Assets` parámetro. Este parámetro es una matriz de archivos de imagen en formato binario codificado en base64.

> **Note**
>
> Las solicitudes programáticas que crean o actualizan un estilo de marca no deben tener un tamaño superior a 2 MB. Es posible que los elementos de su solicitud hagan que supere este límite. En ese caso, divídala en varias solicitudes `UpdateManagedLoginBranding` para grupos de parámetros que no superen el tamaño máximo de la solicitud. Estas solicitudes no dan como resultado que los parámetros no especificados se establezcan de forma predeterminada, por lo que puede enviar solicitudes parciales sin que ello afecte a la configuración existente.

Algunos recursos tienen limitaciones en cuanto a los tipos de archivos que puede enviar.

Activo	Extensiones de archivo aceptadas
FAVICON_ICO	ico
FAVICON_SVG	svg
EMAIL_GRAPHIC	png, svg, jpeg
SMS_GRAPHIC	png, svg, jpeg
AUTH_APP_GRAPHIC	png, svg, jpeg
PASSWORD_GRAPHIC	png, svg, jpeg
PASSKEY_GRAPHIC	png, svg, jpeg
PAGE_HEADER_LOGO	png, svg, jpeg
PAGE_HEADER_BACKGROUND	png, svg, jpeg
PAGE_FOOTER_LOGO	png, svg, jpeg
PAGE_FOOTER_BACKGROUND	png, svg, jpeg
PAGE_BACKGROUND	png, svg, jpeg
FORM_BACKGROUND	png, svg, jpeg
FORM_LOGO	png, svg, jpeg
IDP_BUTTON_ICON	ico, svg

Los archivos del tipo SVG admiten los siguientes atributos y elementos.

Attributes

```
accent-height, accumulate, additivive, alignment-baseline, ascent, attributename,
 attributetype, azimuth, basefrequency, baseline-shift, begin, bias, by, class,
 clip, clip-path, clip-rule, color, color-interpolation, color-interpolation-
filters, color-profile, color-rendering, cx, cy, d, dx, dy, diffuseconstant,
 direction, display, divisor, dur, edgemode, elevation, end, fill, fill-opacity,
```

```
 fill-rule, filter, filterunits, flood-color, flood-opacity, font-family, font-
size, font-size-adjust, font-stretch, font-style, font-variant, font-weight, fx,
 fy, g1, g2, glyph-name, glyphref, gradientunits, gradienttransform, height, href,
 id, image-rendering, in, in2, k, k1, k2, k3, k4, kerning, keypoints, keysplines,
 keytimes, lang, lengthadjust, letter-spacing, kernelmatrix, kernelunitlength,
 lighting-color, local, marker-end, marker-mid, marker-start, markerheight,
 markerunits, markerwidth, maskcontentunits, maskunits, max, mask, media,
 method, mode, min, name, numoctaves, offset, operator, opacity, order, orient,
 orientation, origin, overflow, paint-order, path, pathlength, patterncontentunits,
 patterntransform, patternunits, points, preservealpha, preserveaspectratio, r,
 rx, ry, radius, refx, refy, repeatcount, repeatdur, restart, result, rotate,
 scale, seed, shape-rendering, specularconstant, specularexponent, spreadmethod,
 stddeviation, stitchtiles, stop-color, stop-opacity, stroke-dasharray, stroke-
dashoffset, stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-opacity,
 stroke, stroke-width, style, surfacescale, tabindex, targetx, targety, transform,
 text-anchor, text-decoration, text-rendering, textlength, type, u1, u2, unicode,
 values, viewbox, visibility, vert-adv-y, vert-origin-x, vert-origin-y, width, word-
spacing, wrap, writing-mode, xchannelselector, ychannelselector, x, x1, x2, xmlns,
 y, y1, y2, z, zoomandpan
```

Elements

```
svg, a, altglyph, altglyphdef, altglyphitem, animatecolor, animatemotion,
 animatetransform, audio, canvas, circle, clippath, defs, desc, ellipse, filter,
 font, g, glyph, glyphref, hkern, image, line, lineargradient, marker, mask,
 metadata, mpath, path, pattern, polygon, polyline, radialgradient, rect, stop,
 style, switch, symbol, text, textpath, title, tref, tspan, video, view, vkern,
 feBlend, feColorMatrix, feComponentTransfer, feComposite, feConvolveMatrix,
 feDiffuseLighting, feDisplacementMap, feDistantLight, feFlood, feFuncA, feFuncB,
 feFuncG, feFuncR, feGaussianBlur, feMerge, feMergeNode, feMorphology, feOffset,
 fePointLight, feSpecularLighting, feSpotLight, feTile, feTurbulence
```

Herramientas para operaciones de creación de marcas de inicio de sesión administrado

Amazon Cognito administra un archivo con el JSON-Schema formato del objeto de configuración de marca del inicio de sesión administrado. A continuación, se explica cómo actualizar mediante programación su estilo de marca.

Cómo actualizar la marca en la API de grupos de usuarios

1. En la consola de Amazon Cognito, cree un estilo de marca de inicio de sesión administrado y predeterminado desde el menú Inicio de sesión administrado de su grupo de usuarios. Asígnelo a un cliente de aplicación.
2. Registre el ID del cliente de aplicación para el que creó el estilo, por ejemplo `1example23456789`.
3. Recupere la configuración del estilo de marca con una solicitud de DescribeManagedLoginBrandingByClientAPI con el valor de. `ReturnMergedResources` `true` A continuación, se muestra un ejemplo de cuerpo de la solicitud .

   ```
   {
      "ClientId": "1example23456789",
      "ReturnMergedResources": true,
      "UserPoolId": "us-east-1_EXAMPLE"
   }
   ```

4. Modifique el resultado de `DescribeManagedLoginBrandingByClient` con sus personalizaciones.

   a. El cuerpo de la respuesta está incluido en un elemento `ManagedLoginBranding` que no forma parte de la sintaxis de las operaciones de creación y actualización. Elimine este nivel superior del objeto JSON.

   b. Para reemplazar las imágenes, sustituya el `Bytes` valor por los datos Base64-encoded binarios de cada archivo de imagen.

   c. Para actualizar la configuración, modifique la salida del objeto `Settings` e inclúyala en su próxima solicitud. Amazon Cognito ignora los valores del objeto `Settings` que no estén en el esquema que usted reciba en la respuesta de la API.

5. Usa el cuerpo de la respuesta actualizada en una UpdateManagedLoginBrandingsolicitud CreateManagedLoginBrandingo. Si el tamaño de la solicitud supera los 2 MB, sepárela en varias solicitudes. Estas operaciones funcionan en un modelo `PATCH` en el que la configuración original permanece inalterada a menos que especifique lo contrario.

Migración de la clásica interfaz de usuario alojada al inicio de sesión gestionado

Si su grupo de usuarios utiliza actualmente la interfaz de usuario alojada clásica, puede migrarla al inicio de sesión gestionado. El inicio de sesión gestionado y la interfaz de usuario alojada clásica son las dos versiones de marca. La versión de marca se establece en el dominio del grupo de usuarios y la elección se aplica a todos los clientes de aplicaciones que utilizan ese dominio, por lo que al migrar se cambian todos los clientes de aplicaciones del dominio al mismo tiempo. El inicio de sesión gestionado incluye un editor visual de marca, un inicio de sesión con clave de paso y funciones de autenticación adicionales. La interfaz de usuario alojada clásica utiliza una marca basada en archivos y está disponible en todos los planes de funciones. Para ver una comparación característica por característica, consulte. the section called “Inicio de sesión administrado”

> **Important**
>
> El inicio de sesión gestionado y su editor de marca requieren el plan de funciones Essentials o Plus. La interfaz de usuario alojada clásica está disponible en todos los planes, incluido el Lite. Confirma tu plan de funciones antes de realizar la migración. Consulte the section called “Inicio de sesión administrado”.

En un nivel superior, mover un grupo de usuarios existente de la interfaz de usuario alojada clásica al inicio de sesión gestionado implica los siguientes pasos. Revise cada paso comparándolo con su configuración actual antes de cambiar:

1. Confirme que su grupo de usuarios tiene un plan de funciones que admita el inicio de sesión gestionado. Consulte the section called “Inicio de sesión administrado”.
2. Configure la versión de marca en el dominio de su grupo de usuarios como inicio de sesión gestionado. La versión de marca es una configuración de dominio, por lo que se aplica a todos los clientes de aplicaciones que utilizan ese dominio.
3. Configura tu marca en el editor visual de marca. El inicio de sesión gestionado utiliza un modelo de marca diferente al de la clásica interfaz de usuario alojada, la personalización de logotipos y CSS basada en archivos, por lo que te recomendamos que compruebes tu marca actual comparándola con el editor de marca al cambiarla. Consulte the section called “Creación de marca y personalización”.

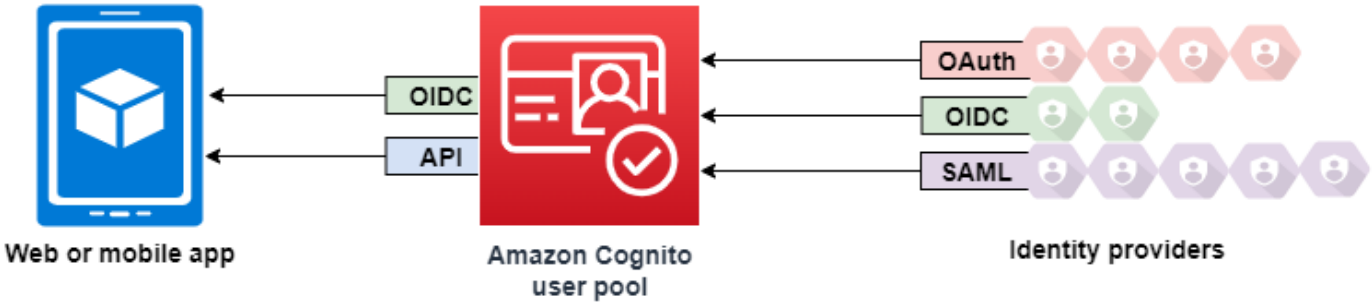
4. Pruebe los flujos de inicio de sesión, registro, autenticación multifactor (MFA) y restablecimiento de contraseñas en las páginas de inicio de sesión gestionadas antes de dirigir el tráfico de producción a ellas.

> **Note**
>
> En esta página se resume la migración a nivel conceptual. Los procedimientos detallados de configuración y marca a los que hace referencia son la fuente autorizada de cada paso.

Autenticación de usuarios con 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.

En este capítulo, se explica cómo se autentican los usuarios: el registro, el inicio de sesión y los flujos de autenticación que admite un grupo de usuarios, incluido el inicio de sesión a través de proveedores de identidad externos. Las tareas relacionadas se tratan en sus propios capítulos: administrar los usuarios de su directorio, personalizar el comportamiento con los AWS Lambda desencadenantes y proteger el grupo con protección contra amenazas.



Temas

- Características
- Planes de características de grupo de usuarios
- Autenticación con grupos de usuarios de Amazon Cognito
- Acceso a los recursos después de iniciar sesión correctamente

Características

Los grupos de usuarios de Amazon Cognito cuentan con las características siguientes.

Sign-up

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, actualice el Self-service registro en el Sign-upmenú de Autenticación en la consola del grupo de usuarios o actualice el valor de una AllowAdminCreateUserOnlysolicitud de API CreateUserPool. UpdateUserPool Para obtener información sobre las características de seguridad que puede configurar en los grupos de usuarios, consulte Protección de grupos de usuarios con protección contra amenazas.

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 AWS credenciales.
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 usted mismo, en función de su propia fuente de datos y esquema, de cualquiera de las siguientes maneras:
   - Añada usuarios directamente en la consola de Amazon Cognito o con la API.
   - Importe usuarios de forma masiva desde un archivo CSV.
   - Ejecute una AWS Lambda función justo a tiempo que busque a cada usuario nuevo en un directorio existente y complete su perfil a partir de esos datos.

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

- Administración de usuarios en el grupo de usuarios
- Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado
- Ejemplos de código para Amazon Cognito Identity Provider mediante AWS SDK

Sign-in

Amazon Cognito puede ser un directorio de usuarios independiente y proveedor de identidades (IdP) para la aplicación. Los usuarios pueden iniciar sesión con páginas de inicio de sesión administrado alojadas por Amazon Cognito o con un servicio de autenticación de usuarios personalizado a través de la API de grupos de usuarios de Amazon Cognito. El nivel de la aplicación que está detrás del frontend personalizado puede autorizar las solicitudes en el backend con cualquiera de los métodos para confirmar las solicitudes legítimas.

Los usuarios pueden configurar e iniciar sesión con nombres de usuario y contraseñas, claves de paso y contraseñas de un solo uso para correos electrónicos y mensajes SMS. Puede ofrecer un inicio de sesión consolidado con directorios de usuarios externos, autenticación multifactor (MFA) después del inicio de sesión, dispositivos recordados de confianza y flujos de autenticación personalizados que diseñe.

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 clientes 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 y educativos 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

- Inicio de sesión en el grupo de usuarios con proveedores de identidad externos
- Vinculación de usuarios federados a un perfil de usuario existente

Machine-to-machine autorización

Algunas sesiones no son una interacción entre humanos y máquinas. 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 autorización de máquina a máquina con los ámbitos de OAuth 2.0, puedes añadir un cliente de aplicación que genere concesiones de credenciales de cliente desde el punto de enlace del token o llamar a la operación `GetClientToken` API, que no requiere un dominio de grupo de usuarios.

Temas relacionados

- Ámbitos, M2M y servidores de recursos

Inicio de sesión administrado

Si no quieres crear una interfaz de usuario, puedes ofrecer a tus usuarios páginas de inicio de sesión gestionadas y personalizadas. El inicio de sesión administrado es un conjunto de páginas web para registrarse, iniciar sesión, emplear la autenticación multifactor (MFA) y restablecer contraseña. Puedes añadir un inicio de sesión gestionado a tu dominio existente o usar un dominio con prefijo en un AWS subdominio.

El inicio de sesión gestionado tiene una alternativa: la clásica interfaz de usuario alojada, una versión de primera generación con un diseño más sencillo y menos funciones. Puede elegir uno u otro para el dominio de un grupo de usuarios; no se utilizan juntos. En la tabla siguiente se comparan las dos opciones.

Característica	Interfaz de usuario alojada clásica	Inicio de sesión administrado
Versión de marca	Interfaz de usuario alojada (clásica)	Inicio de sesión administrado

Característica	Interfaz de usuario alojada clásica	Inicio de sesión administrado
Personalización	File-based marca (imagen del logotipo y un archivo de valores CSS)	Editor visual de marca en la consola Amazon Cognito
Planes de características	Todos los planes, incluido Lite	Essentials y Plus
Inicio de sesión con clave de paso	No disponible	Disponible

> **Note**
>
> La clásica interfaz de usuario alojada está disponible en todos los planes de funciones, incluido Lite. El inicio de sesión gestionado y su editor de marca requieren el plan de funciones Essentials o Plus. Para obtener más información, consulte Planes de características de grupo de usuarios.

Para configurar el inicio de sesión gestionado, complete estos requisitos previos en orden:

1. Cree un grupo de usuarios. Consulte Creación de una nueva aplicación en la consola de Amazon Cognito.
2. Agregue un dominio de grupo de usuarios (un dominio de prefijo o un dominio personalizado). Consulte Configuración de un dominio del grupo de usuarios.
3. Cree un cliente de aplicación y establezca su versión de marca y la URL de retorno. Consulte Inicio de sesión administrado de grupos de usuarios.

Temas relacionados

- Inicio de sesión administrado de grupos de usuarios
- Configuración de un dominio del grupo de usuarios

Protección contra amenazas

Proteja a sus usuarios contra la adivinación de contraseñas, la apropiación de cuentas y el tráfico malicioso con la autenticación multifactor (MFA), la autenticación adaptativa y las ACL web. AWS WAF Estas protecciones están incluidas en. Protección de grupos de usuarios con protección contra amenazas

Personalizar la experiencia del usuario

En la mayoría de las etapas del registro, el inicio de sesión o la actualización del perfil, puede personalizar la forma en que Amazon Cognito gestiona la solicitud con activadores de Lambda y flujos de autenticación personalizados, y dar al inicio de sesión gestionado un aspecto familiar. Consulte Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda.

Monitoreo y análisis

Los grupos de usuarios de Amazon Cognito registran las solicitudes de API AWS CloudTrail, publican métricas de rendimiento en Amazon CloudWatch Logs y supervisan la entrega de mensajes y el volumen de solicitudes. Consulte Inicio de sesión en Amazon Cognito AWS CloudTrail.

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

- Acceso Servicios de AWS usar un grupo de identidades después de iniciar sesión
- Grupos de identidades de Amazon Cognito

Planes de características de grupo de usuarios

Comprender el costo es un paso crucial para prepararse para implementar la autenticación de grupos de usuarios de Amazon Cognito. Amazon Cognito tiene planes de características para grupos de usuarios. Cada plan tiene un conjunto de características y un costo mensual por usuario activo. Cada plan de características desbloquea el acceso a más características que el anterior.

Los grupos de usuarios tienen una variedad de características que puede activar y desactivar. Por ejemplo, puedes activar la autenticación multifactor (MFA) y desactivar el inicio de sesión con proveedores de identidad externos (). IdPs Algunos cambios requieren que cambie de plan de características. Las siguientes características de su grupo de usuarios determinan el coste que se le AWS factura mensualmente por su uso.

- Las características que elija
- Las solicitudes por segundo que la aplicación realiza a la API de grupos de usuarios
- El número de usuarios con actividad de autenticación, actualización o consulta en un mes, también denominados usuarios activos mensuales o MAU
- El número de usuarios activos mensuales de SAML 2.0 u OpenID Connect (OIDC) de terceros IdPs
- La cantidad de clientes de aplicaciones y grupos de usuarios que utilizan credenciales de cliente para la autorización máquina a máquina

Para obtener la información más actualizada sobre los precios de los grupos de usuarios, consulte los Precios de Amazon Cognito.

Feature-plan las selecciones se aplican a un grupo de usuarios. Los distintos grupos de usuarios de una misma Cuenta de AWS pueden tener distintas selecciones de planes. No puede aplicar planes de características independientes a los clientes de aplicaciones de un grupo de usuarios. La selección predeterminada del plan para los nuevos grupos de usuarios es Essentials.

Puede cambiar de un plan de características a otro en cualquier momento para adaptarse a los requisitos de sus aplicaciones. Algunos cambios entre planes requieren la desactivación de las características activas. Para obtener más información, consulte Desactivación de las características para cambiar los planes de características.

Planes de características de grupo de usuarios

Lite

Lite es un plan de características de bajo costo para grupos de usuarios con un número menor de usuarios activos mensuales. Este plan es suficiente para los directorios de usuarios con características de autenticación básicas. Incluye funciones de inicio de sesión y la clásica interfaz de usuario alojada, una predecesora más delgada y menos personalizable del inicio de sesión gestionado. Muchas de las características más recientes, como la personalización del token de acceso y la autenticación con clave de acceso, no están incluidas en el plan Lite.

Essentials

Essentials cuenta con todas las características de autenticación de grupos de usuarios más recientes. Este plan añade nuevas opciones a sus aplicaciones, tanto si sus páginas de inicio de sesión son de inicio de sesión administrado como si están personalizadas. Essentials tiene características de autenticación avanzadas, como el inicio de sesión basado en opciones y la MFA por correo electrónico.

Plus

Plus incluye todo lo que hay en el plan Essentials y añade características de seguridad avanzadas que protegen a sus usuarios. Supervise las solicitudes de inicio de sesión, registro y administración de contraseñas de los usuarios para detectar indicadores de compromiso. Por ejemplo, los grupos de usuarios pueden detectar si los usuarios inician sesión desde una ubicación inesperada o si utilizan una contraseña que ha sido parte de una infracción pública.

Los grupos de usuarios del plan Plus generan registros con los detalles de la actividad de los usuarios y las evaluaciones de riesgos. Puede aplicar su propio análisis de uso y seguridad a estos registros cuando los exporte a servicios externos.

> **Note**
>
> Anteriormente, algunas características del grupo de usuarios se incluían en una estructura de precios de características de seguridad avanzadas. Las características que se incluían en esta estructura ahora se incluyen en el plan Essentials o Plus.

Temas

- Selección de un plan de características

- Características por plan
- Características del plan Essentials
- Plan de características Plus
- Desactivación de las características para cambiar los planes de características

Selección de un plan de características

Consola de administración de AWS

Cómo elegir un plan de características

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 un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Seleccione el menú Configuración y consulte la pestaña Planes de características.
5. Revise las características disponibles en los planes Lite, Esssentials y Plus.
6. Para cambiar el plan, seleccione Cambiar a Essentials o Cambiar a Plus. Para cambiar al plan Lite, seleccione Otros planes y, luego, Comparar con Lite.
7. En la siguiente pantalla, revise su elección y seleccione Confirmar.

CLI/API/SDK

Las UpdateUserPooloperaciones CreateUserPooly configuran su plan de funciones en el `UserPoolTier` parámetro. Si no especifica un valor para `UserPoolTier`, el grupo de usuarios se pone de forma predeterminada en el plan `Essentials`. Si configura `AdvancedSecurityMode` en `AUDIT` o `ENFORCED`, el nivel de su grupo de usuarios debe ser `PLUS` y ponerse en `PLUS` de forma predeterminada cuando no se especifique nada.

Consulte los ejemplos CreateUserPool para ver la sintaxis. Consulte la sección Vea también en CreateUserPool para ver los enlaces a esta función en AWS los SDK de diversos lenguajes de programación.

```
"UserPoolTier": "PLUS"
```

En el AWS CLI, esta opción es `--user-pool-tier` argumento.

```
--user-pool-tier PLUS
```

Consulte create-user-pool y update-user-pool en la referencia de comandos para obtener más información. AWS CLI

Características por plan

Características y planes en los grupos de usuarios

Característica	Description (Descripción)	Plan de características
Protección contra contraseñas no seguras	Comprobar las contraseñas de texto simple para ver si hay indicios de que hayan sido comprometidas durante el tiempo de ejecución	Plus
Protección contra intentos de inicio de sesión malintenc ionados	Comprobar las propiedad es de la sesión en busca de indicadores de compromiso durante el tiempo de ejecución	Plus
Registro y análisis de la actividad del usuario	Generar registros sobre las propiedades de las sesiones de autenticación de los usuarios y de las puntuacio nes de riesgo	Plus
Exportación de registros de actividad de los usuarios	Envíe los registros de sesión y riesgo del usuario a un sitio externo Servicio de AWS	Plus
Personalización de las páginas de inicio de sesión administrado con un editor visual	Usar un editor visual en la consola de Amazon Cognito para aplicar la marca y el estilo a las páginas de inicio de sesión administrado	Essentials + Plus

Característica	Description (Descripción)	Plan de características
MFA con códigos de correo electrónico de un solo uso	Solicitar o exigir a los usuarios locales que proporcionen un factor de inicio de sesión adicional mediante un mensaje de correo electróni co tras la autenticación del nombre de usuario	Essentials + Plus
Personalización de los ámbitos y las reclamaciones de los tokens de acceso en tiempo de ejecución	Usar un desencadenador de Lambda para ampliar las capacidades de autorización de los tokens de acceso al grupo de usuarios	Essentials + Plus
Inicio de sesión sin contraseña con códigos de un solo uso	Permitir que los usuarios reciban una contraseña de un solo uso por correo electrónico o SMS como primer factor de autenticación	Essentials + Plus
Inicio de sesión con clave de acceso con autenticadores FIDO2 de hardware o software	Permitir a los usuarios utilizar una clave criptográfica almacenada en un autentica dor FIDO2 como primer factor de autenticación	Essentials + Plus
Sign-up e iniciar sesión	Realizar operaciones de autenticación y permitir que los nuevos usuarios se registren para obtener una cuenta en su aplicación.	Lite + Essentials + Plus

Característica	Description (Descripción)	Plan de características
Grupos de usuarios	Crear agrupaciones lógicas de usuarios y asignar roles de IAM predeterminados para las operaciones del grupo de identidades.	Lite + Essentials + Plus
Sign-in con proveedores de redes sociales, SAML y OIDC	Ofrecer a los usuarios la opción de iniciar sesión directamente o con su proveedor preferido.	Lite + Essentials + Plus
OAuth 2. 0/OIDC servidor de autorización	Actuar como emisor de OIDC.	Lite + Essentials + Plus
Páginas de inicio de sesión	Una colección alojada de páginas web para la autentica ción. El inicio de sesión administrado está disponibl e en los niveles Essentials y Plus. La interfaz de usuario alojada clásica está disponibl e en todos los niveles de características.	Lite + Essentials + Plus
Autenticación con contraseñ a, personalizada, con token de actualización y SRP	Solicitar a los usuarios un nombre de usuario y contraseña en la aplicación.	Lite + Essentials + Plus
Machine-to-machine (M2M) con credenciales de cliente	Emitir tokens de acceso para la autorización de entidades no humanas.	Lite + Essentials + Plus
Autorización API con servidore s de recursos	Emitir tokens de acceso con ámbitos personalizados que autoricen el acceso a sistemas externos.	Lite + Essentials + Plus

Característica	Description (Descripción)	Plan de características
Importación de usuarios	Configurar los trabajos de importación desde archivos CSV y realizar la migración justo a tiempo para los usuarios a medida que inician sesión	Lite + Essentials + Plus
MFA con aplicaciones de autenticación y códigos SMS de un solo uso	Solicitar o exigir a los usuarios locales que proporcionen un factor de inicio de sesión adicional con aplicación de autenticación o mensaje SMS tras la autenticación del nombre de usuario	Lite + Essentials + Plus
Personalización de los ámbitos y las reclamaciones de los tokens de ID en tiempo de ejecución	Usar un desencadenador de Lambda para ampliar las capacidades de autenticación de los tokens de identidad (ID) del grupo de usuarios	Lite + Essentials + Plus
Acciones de tiempo de ejecución personalizadas con desencadenadores de Lambda	Personalizar el proceso de inicio de sesión en tiempo de ejecución con funciones de Lambda que realicen acciones externas e influyan en la autenticación	Lite + Essentials + Plus
Personalización de las páginas de inicio de sesión administrado con CSS	Descargar una plantilla CSS y cambiar algunos estilos en sus páginas de inicio de sesión administrado	Lite + Essentials + Plus

Características del plan Essentials

El plan de características Essentials incluye la mayoría de las mejores y más recientes características de los grupos de usuarios de Amazon Cognito. Al cambiar del plan Lite al Essentials, obtendrá nuevas características para sus páginas de inicio de sesión administrado, una autenticación multifactor con contraseñas de un solo uso para los mensajes de correo electrónico, una política de contraseñas mejorada y tokens de acceso personalizados. Para mantenerse al día con las nuevas características del grupo de usuarios, elija el plan Essentials para sus grupos de usuarios.

En las siguientes secciones se presenta un breve resumen de las características que puede añadir a su aplicación con el plan Essentials. Para obtener información detallada, consulte las siguientes páginas.

Recursos adicionales

- Personalización del token de acceso: Desencadenador de Lambda anterior a la generación del token
- MFA de correo electrónico: MFA con mensajes SMS y correo electrónico
- Historial de contraseñas: Contraseñas, recuperación de contraseñas y políticas de contraseñas
- IU mejorada: Aplicación de la creación de marca en las páginas de inicio de sesión administrado

Temas

- Personalización del token de acceso
- MFA de correo electrónico
- Prevención de la reutilización de contraseñas
- Servidor de autorización e inicio de sesión administrado y alojado
- Choice-based authentication

Personalización del token de acceso

Los tokens de acceso del grupo de usuarios otorgan permisos a las aplicaciones para acceder a una API, para recuperar los atributos de usuario del punto de conexión de userInfo o para establecer la pertenencia a un grupo de un sistema externo. En situaciones avanzadas, puede que desee añadir al token de acceso predeterminado datos del directorio del grupo de usuarios con parámetros temporales adicionales que la aplicación determine en tiempo de ejecución. Es posible, por ejemplo,

que desee verificar los permisos de API de un usuario con Amazon Verified Permissions y ajustar debidamente los ámbitos en el token de acceso.

El plan Essentials se suma a las funciones ya existentes de un desencadenador Antes de la generación del token. Con los planes de nivel inferior, puede personalizar los tokens de ID con reclamaciones, roles y pertenencia a grupos adicionales. Essentials añade nuevas versiones del evento de desencadenador que personalizan las notificaciones, los roles, la pertenencia a grupos y los ámbitos de los tokens de acceso. La personalización del token de acceso está disponible para la concesión de credenciales de cliente de máquina a máquina (M2M) con la versión 3 de eventos.

Personalización de tokens de acceso

1. Seleccione el plan de características Essentials o Plus.
2. Cree una función de Lambda para el desencadenador. Para usar nuestra función de ejemplo, configúrela para Node.js.
3. Rellene la función de Lambda con nuestro código de ejemplo o cree el suyo propio. La función debe procesar un objeto de solicitud de Amazon Cognito y devolver los cambios que desee incluir.
4. Asigne la nueva función como un desencadenador Antes de la generación del token en la versión 2 o 3. Los eventos de la versión 2 personalizan los tokens de acceso para las identidades de los usuarios. La versión 3 personaliza los tokens de acceso para las identidades de usuario y máquina.

Más información

- Personalización del token de acceso
- How to customize access tokens in Amazon Cognito user pools

MFA de correo electrónico

Los grupos de usuarios de Amazon Cognito se pueden configurar para utilizar el correo electrónico como segundo factor en la autenticación multifactor (MFA). Con la MFA de correo electrónico, Amazon Cognito puede enviar a los usuarios un correo electrónico con un código de verificación que deben introducir para completar el proceso de autenticación. Esto añade una importante capa adicional de seguridad al flujo de inicio de sesión de los usuarios. Para habilitar la MFA basada en correo electrónico, el grupo de usuarios debe estar configurado para usar la configuración de

envío de correo electrónico de Amazon SES en lugar de la configuración de correo electrónico predeterminada.

Cuando el usuario selecciona MFA por mensaje de correo electrónico, Amazon Cognito envía un código de verificación de un solo uso a la dirección de correo electrónico registrada del usuario cada vez que intenta iniciar sesión. A continuación, el usuario debe devolver ese código al grupo de usuarios para completar el flujo de autenticación y obtener acceso. Esto garantiza que, incluso si el nombre de usuario y la contraseña de un usuario están comprometidos, este debe proporcionar un factor adicional (el código enviado por correo electrónico) para poder acceder a los recursos de la aplicación.

Para obtener más información, consulte MFA con mensajes SMS y correo electrónico. A continuación, se ofrece información general sobre cómo configurar el grupo de usuarios y los usuarios para la MFA de correo electrónico.

Cómo configurar la MFA de correo electrónico en la consola de Amazon Cognito

1. Seleccione el plan de características Essentials o Plus.
2. En el Sign-inmenú de su grupo de usuarios, edite Multi-factor la autenticación.
3. Elija el nivel de Cumplimiento de MFA que desee configurar. Con Requerir MFA, los usuarios de la API reciben automáticamente un desafío para configurar, confirmar e iniciar sesión mediante una MFA. En los grupos de usuarios que requieren MFA, el inicio de sesión administrado les solicita que elijan y configuren un factor MFA. Con la opción MFA opcional, la aplicación debe ofrecer a los usuarios la posibilidad de configurar la MFA y establecer las preferencias del usuario para la MFA por correo electrónico.
4. En Métodos de MFA, seleccione Mensaje de correo electrónico como una de las opciones.

Más información

- MFA con mensajes SMS y correo electrónico

Prevención de la reutilización de contraseñas

De forma predeterminada, la política de contraseñas de los grupos de usuarios de Amazon Cognito establece los requisitos de longitud y tipo de caracteres de las contraseñas y su caducidad temporal. El plan Essentials añade la capacidad de aplicar el historial de contraseñas. Cuando un usuario intenta restablecer la contraseña, el grupo de usuarios puede impedir que la configure con

una contraseña anterior. Para obtener más información acerca de la configuración de la política de contraseñas, consulte Adición de requisitos de contraseña para los grupos de usuarios. A continuación, se ofrece información general sobre cómo configurar un grupo de usuarios con una política de historial de contraseñas.

Cómo configurar el historial de contraseñas en la consola de Amazon Cognito

1. Seleccione el plan de características Essentials o Plus.
2. En el menú Métodos de autenticación de su grupo de usuarios, busque la Política de contraseñas y seleccione Editar.
3. Configure otras opciones disponibles y establezca un valor para Impedir el uso de contraseñas utilizadas anteriormente.

Más información

- Contraseñas, recuperación de contraseñas y políticas de contraseñas

Servidor de autorización e inicio de sesión administrado y alojado

Los grupos de usuarios de Amazon Cognito tienen páginas web opcionales que admiten las siguientes funciones: un IdP de OpenID Connect (OIDC), un proveedor de servicios o una parte de confianza de un IdPs tercero y páginas públicas interactivas para los usuarios para registrarse e iniciar sesión. Estas páginas se denominan colectivamente inicio de sesión administrado. Cuando elige un dominio para su grupo de usuarios, Amazon Cognito activa automáticamente estas páginas. Mientras que el plan Lite tiene la interfaz de usuario alojada, el plan Essentials abre esta versión avanzada de las páginas de registro e inicio de sesión.

Las páginas de inicio de sesión administrado tienen una interfaz limpia y actualizada, con más características y opciones para personalizar la marca y los estilos. El plan Essentials es el nivel de plan más bajo para poder desbloquear el acceso al inicio de sesión administrado.

Cómo configurar el inicio de sesión administrado en la consola de Amazon Cognito.

1. En el menú Configuración, seleccione el plan de características Essentials o Plus.
2. En el menú Dominio, asigne un dominio a su grupo de usuarios y seleccione una versión de marca del inicio de sesión administrado.

3. En el menú de inicio de sesión administrado, en la pestaña Estilos, seleccione Crear un estilo y asígnelo a un cliente de aplicación o cree un nuevo cliente de aplicación.

Más información

- Inicio de sesión administrado de grupos de usuarios

Choice-based authentication

El nivel Essentials introduce un nuevo flujo de autenticación para las operaciones de autenticación en las operaciones de interfaz de usuario y API mejoradas. Este flujo se basa en la autenticación basada en las elecciones. SDK-based Choice-based la autenticación es un método en el que la autenticación de los usuarios no comienza con una declaración de un método de inicio de sesión en la aplicación, sino con una consulta de los posibles métodos de inicio de sesión seguida de una elección. Puede configurar su grupo de usuarios para que admita la autenticación basada en opciones y desbloquee la autenticación con nombre de usuario y contraseña, sin contraseña y con clave de acceso. En la API, este es el flujo `USER_AUTH`.

Configuración de la autenticación basada en opciones en la consola de Amazon Cognito

1. Seleccione el plan de características Essentials o Plus.
2. En el Sign-inmenú de tu grupo de usuarios, edita las opciones para iniciar sesión en función de tus elecciones. Seleccione y configure los métodos de autenticación que desee habilitar en la autenticación basada en opciones.
3. En el menú Métodos de autenticación de su grupo de usuarios, edite la configuración de las operaciones de inicio de sesión.

Más información

- Autenticación con grupos de usuarios de Amazon Cognito

Plan de características Plus

El plan de características Plus incluye características de seguridad avanzadas para los grupos de usuarios de Amazon Cognito. Estas características registran y analizan el contexto del usuario en tiempo de ejecución para detectar posibles problemas de seguridad en los dispositivos, las

ubicaciones, los datos de solicitud y las contraseñas. Luego, mitigan los posibles riesgos con respuestas automáticas que bloquean o añaden medidas de seguridad a las cuentas de los usuarios. También puede exportar sus registros de seguridad a Amazon S3, Amazon Data Firehose o Amazon CloudWatch Logs para analizarlos más a fondo.

Al cambiar del plan Essentials al plan Plus, obtiene todas las características de Essentials y las características adicionales correspondientes. Estas incluyen el conjunto de opciones de seguridad de protección contra amenazas, también conocidas como características de seguridad avanzadas. A fin de configurar sus grupos de usuarios para que se adapten automáticamente a las amenazas en su interfaz de autenticación, elija el plan Plus para sus grupos de usuarios.

En las siguientes secciones se presenta un breve resumen de las características que puede añadir a su aplicación con el plan Plus. Para obtener información detallada, consulte las siguientes páginas.

Recursos adicionales

- Autenticación adaptativa: Uso de la autenticación flexible
- Credenciales comprometidas: Uso de la detección de credenciales comprometidas
- Exportación de registros; Exportación de registros de grupos de usuarios de Amazon Cognito

Temas

- Protección contra amenazas: autenticación flexible
- Protección contra amenazas: detección de credenciales comprometidas
- Protección contra amenazas: registro de la actividad de los usuarios

Protección contra amenazas: autenticación flexible

El plan Plus incluye una característica de autenticación flexible. Al activar esta característica, su grupo de usuarios realiza una evaluación de riesgos de cada sesión de autenticación de usuarios. A partir de las clasificaciones de riesgo resultantes, puede bloquear la autenticación o solicitar la MFA para los usuarios que inicien sesión con un nivel de riesgo superior al umbral que usted determine. Con la autenticación flexible, su grupo de usuarios y su aplicación bloquean o configuran automáticamente la MFA para los usuarios cuyas cuentas sospecha que están siendo atacadas. También puede proporcionar comentarios sobre las calificaciones de riesgo de su grupo de usuarios para ajustar las calificaciones futuras.

Cómo configurar la autenticación flexible en la consola de Amazon Cognito

1. Seleccione el plan de características Plus.
2. En el menú Protección contra amenazas de su grupo de usuarios, edite la autenticación estándar y personalizada en Protección contra amenazas.
3. Establezca el modo de aplicación para la autenticación estándar o personalizada en Full-function.
4. En Autenticación flexible, configure las respuestas automáticas al riesgo para los distintos niveles de riesgo.

Más información

- Uso de la autenticación flexible
- Recopilación de datos para la protección contra amenazas en las aplicaciones

Protección contra amenazas: detección de credenciales comprometidas

El plan Plus incluye una característica de detección de credenciales comprometidas. Esta característica protege contra el uso de contraseñas inseguras y la amenaza de acceso no deseado a las aplicaciones que esta práctica genera. Si permite que sus usuarios inicien sesión con un nombre de usuario y una contraseña, es posible que reutilicen una contraseña que hayan utilizado en otro lugar. Es posible que esa contraseña se haya filtrado o que simplemente se haya adivinado por tratarse de algo frecuente. Con la detección de credenciales comprometidas, su grupo de usuarios lee las contraseñas que envían sus usuarios y las compara con las bases de datos de contraseñas. Si la operación da como resultado la decisión de que es probable que la contraseña esté comprometida, puede configurar su grupo de usuarios para bloquear el inicio de sesión y, a continuación, iniciar el restablecimiento de la contraseña del usuario de la aplicación.

Compromised-credentials la detección puede reaccionar ante contraseñas inseguras cuando se registran nuevos usuarios, cuando los usuarios existentes inician sesión y cuando los usuarios intentan restablecer sus contraseñas. Con esta característica, su grupo de usuarios puede impedir el inicio de sesión con contraseñas inseguras dondequiera que los usuarios las introduzcan o generar una advertencia al respecto.

Configuración de la detección de credenciales comprometidas en la consola de Amazon Cognito

1. Seleccione el plan de características Plus.

2. En el menú Protección contra amenazas de su grupo de usuarios, edite la autenticación estándar y personalizada en Protección contra amenazas.
3. Defina el modo de aplicación de la autenticación estándar o personalizada en. Full-function
4. En Credenciales comprometidas, configure los tipos de operaciones de autenticación que desee comprobar y la respuesta automática que desee obtener de su grupo de usuarios.

Más información

- Uso de la detección de credenciales comprometidas

Protección contra amenazas: registro de la actividad de los usuarios

El plan Plus añade una característica de registro que proporciona análisis de seguridad y detalles de los intentos de autenticación de los usuarios. Puede ver evaluaciones de riesgos, direcciones IP de usuarios, agentes de usuario y otra información sobre el dispositivo que se conectó a su aplicación. Puede utilizar esta información con las características de protección contra amenazas integradas, o puede analizar los registros de sus propios sistemas y tomar las medidas adecuadas. Puede exportar los registros de Threat Protection a Amazon S3, CloudWatch Logs o Amazon DynamoDB.

Configuración del registro de actividad de usuarios en la consola de Amazon Cognito

1. Seleccione el plan de características Plus.
2. En el menú Protección contra amenazas de su grupo de usuarios, edite la autenticación estándar y personalizada en Protección contra amenazas.
3. Establezca el modo de aplicación para la autenticación estándar o personalizada en. Audit-only Esta es la configuración mínima para los registros. También puede activarlo en Full-functionmodo y configurar otras funciones de protección contra amenazas.
4. Para exportar los registros a otro Servicio de AWS para su análisis por parte de terceros, vaya al menú de transmisión de registros de su grupo de usuarios y configure un destino de exportación.

Más información

- Exportación de eventos de autenticación de usuarios
- Exportación de registros de grupos de usuarios de Amazon Cognito

Desactivación de las características para cambiar los planes de características

Los planes de características añaden opciones de configuración al grupo de usuarios. Puede configurar y utilizar estas características solo cuando el plan de características relacionado esté activo. Por ejemplo, puede configurar la personalización del token de acceso en los planes Plus y Essentials, pero no en el plan Lite. Para desactivar estas características, debe desactivar todos los componentes activos. La opción Cambiar a del menú Configuración de la consola de Amazon Cognito le informa de las características que debe desactivar antes de poder cambiar su plan. En este capítulo, aprenderá cuáles son los cambios que la desactivación introduce en la configuración del grupo de usuarios y cómo desactivar estas características de forma individual.

Personalización del token de acceso

Para cambiar a un plan que no incluya la personalización del token de acceso, debe eliminar el desencadenador de Lambda Antes de la generación del token de su grupo de usuarios. Para añadir un nuevo desencadenador previo a la generación del token sin tener que acceder a la personalización del token, asigne una nueva función al desencadenador y configúrela para los eventos de `V1_0`. Estos eventos de desencadenador de la versión uno solo pueden procesar los cambios en los tokens de identificación.

Para desactivar manualmente la personalización del token de acceso, elimine el desencadenador previo a la generación del token y añada un desencadenador nuevo de la versión uno.

Protección contra amenazas

Para cambiar a un plan sin protección contra amenazas, desactive todas las características del menú Protección contra amenazas de su grupo de usuarios.

Exportación de registros

Para cambiar a un plan sin exportación de registros, desactívelo desde el menú Secuencia de registro de su grupo de usuarios. El grupo de usuarios dejará de generar registros de actividad de los usuarios locales y exportados. También puedes enviar una solicitud de SetLogDeliveryConfigurationAPI que elimine cualquier configuración cuyo `EventSource` valor sea de`UserActivity`.

MFA de correo electrónico

Para cambiarte a un plan sin MFA por correo electrónico, ve al Sign-inmenú de tu grupo de usuarios. Edite Multi-factor la autenticación y deseleccione el mensaje de correo electrónico como uno de los métodos de MFA disponibles.

Autenticación con grupos de usuarios de Amazon Cognito

Amazon Cognito incluye varios métodos para autenticar a los usuarios. Los usuarios pueden iniciar sesión con WebAuthn contraseñas y claves de acceso. Amazon Cognito puede enviarles una contraseña de un solo uso en un mensaje de correo electrónico o SMS. Puede implementar funciones de Lambda que organicen su propia secuencia de desafíos y respuestas. Son flujos de autenticación. En los flujos de autenticación, los usuarios proporcionan un secreto y Amazon Cognito lo verifica; luego, emite tokens web JSON (JWT) para que las aplicaciones los procesen con las bibliotecas de OIDC. En este capítulo se describe cómo configurar los grupos de usuarios y los clientes de aplicaciones para varios flujos de autenticación en distintos entornos de aplicaciones. En él se describen las opciones para el uso de las páginas de inicio de sesión alojadas del inicio de sesión gestionado y para crear una lógica y una interfaz propias en un AWS SDK.

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.

A fin de verificar la identidad de los usuarios, Amazon Cognito admite flujos de autenticación que incorporan tipos de desafíos además de contraseñas, como contraseñas de un solo uso enviadas por correo electrónico y SMS, y claves de acceso.

Antes de configurar los detalles, elija el enfoque de autenticación que mejor se adapte a su aplicación. La siguiente ayuda para la toma de decisiones resume los dos puntos de entrada y, dentro de una aplicación personalizada, los dos flujos de inicio de sesión iniciales. Cada opción enlaza con su sección detallada.

¿Desea que Amazon Cognito aloje las páginas de inicio de sesión o cree su propia interfaz?

Si desea utilizar la ruta de menor esfuerzo, utilice el inicio de sesión administrado. Amazon Cognito aloja las páginas de inicio, cierre y restablecimiento de contraseñas, y la aplicación procesa el resultado con una biblioteca de terceros de OpenID Connect (OIDC). El inicio de

sesión gestionado presenta automáticamente los métodos de autenticación que permite la configuración del grupo de usuarios y del cliente de la aplicación.

Si quieres crear tu propia interfaz de usuario y controlar la lógica de inicio de sesión, usa un AWS SDK. Tu aplicación llama directamente a la API de grupos de usuarios y tú implementas la lógica de desafío-respuesta.

En una aplicación personalizada, ¿se recopila el método de inicio de sesión del usuario o se declara por adelantado?

Elige la autenticación basada en elecciones (el `USER_AUTH` flujo) si quieres ofrecer a los usuarios una lista de los métodos de inicio de sesión disponibles y permitirles seleccionar uno. Este flujo es la única forma de ofrecer un inicio de sesión sin contraseña y con clave de paso.

Elige la autenticación basada en el cliente cuando tu aplicación ya sepa cómo va a iniciar sesión el usuario y declare el flujo por adelantado, por ejemplo. `USER_SRP_AUTH` Client-basedla autenticación es la única forma de utilizar los flujos personalizados de autenticación y actualización de los tokens.

Requisitos previos para la autenticación basada en elecciones

Choice-based la autenticación se aplica tanto al inicio de sesión gestionado como a los SDK. AWS Para usarla, añádala `ALLOW_USER_AUTH` a los flujos de autenticación permitidos del cliente de la aplicación. Los métodos sin contraseña y con clave de paso también requieren un dominio de inicio de sesión administrado y un plan de funciones superior al nivel Lite (Essentials o Plus).

Temas

- Application-specific configuración con clientes de aplicaciones
- Implementación de flujos de autenticación
- Factores que debe tener en cuenta sobre la autenticación con grupos de usuarios
- Un ejemplo de sesión de autenticación
- Configuración de los métodos de autenticación para el inicio de sesión administrado
- Administre los métodos de autenticación en AWS SDK
- Flujos de autenticación
- Modelos de autorización para la autenticación de API y SDK
- Inicio de sesión en el grupo de usuarios con proveedores de identidad externos

Application-specific configuración con clientes de aplicaciones

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 que firme las solicitudes de API autenticadas con AWS credenciales, debes proteger tus credenciales para que no las inspeccione el usuario.

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.

Puede aplicar ajustes para las siguientes características del grupo de usuarios en el cliente de aplicación:

1. Análisis
2. Inicio de sesión gestionado IdPs, tipos de concesiones, URL de devolución de llamadas y personalización
3. Servidores de recursos y ámbitos personalizados
4. Protección contra amenazas
5. Permisos de lectura y escritura de atributos
6. Caducidad y revocación de los tokens
7. Flujos de autenticación

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. La aplicación puede ejecutarse como un daemon o un script de shell en tu servidor back-end.

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.

Cada cliente de la aplicación puede tener hasta dos secretos a la vez, lo que permite la rotación de secretos sin tiempo de inactividad. Al crear un cliente de aplicación, puede dejar que Amazon Cognito genere un valor secreto o proporcionar su propio valor secreto personalizado. No puede cambiar secretos del cliente después de crear una aplicación. Puede añadir un segundo secreto con la operación de AddUserPoolClientSecretAPI para rotar los secretos. Al añadir un secreto, puede dejar que Amazon Cognito genere un valor secreto o proporcionar su propio valor secreto personalizado. Para eliminar un secreto, utilice la operación de DeleteUserPoolClientSecretAPI. No puedes eliminar el único secreto asociado a un cliente de aplicación. También puede eliminar una aplicación para bloquear el acceso de aplicaciones que utilizan el ID de cliente de dicha aplicación.

> **Note**
>
> La consola de Amazon Cognito crea clientes de aplicaciones con secretos de cliente al seleccionar la aplicación web tradicional y Machine-to-machine las opciones de aplicación para el tipo de aplicación. Elija una de estas opciones para generar un secreto de cliente o cree el cliente mediante programación con CreateUserPoolClienty configúrelo como. `GenerateSecret` `true`

> **Important**
>
> No puedes cambiar si un cliente de aplicación tiene un secreto de cliente después de crear el cliente de aplicación. La UpdateUserPoolClientoperación no puede añadir ni eliminar un secreto de cliente. Para cambiar esta configuración, cree un nuevo cliente de aplicación. No es necesario volver a crear el grupo de usuarios y los demás clientes de la aplicación no se ven afectados.

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 Descripción del token de identidad (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 Descripción 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 Tokens de actualización.

Puede establecer la caducidad de estos tokens para cada cliente de aplicación desde el menú Clientes de aplicación 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 llamadas permitidas

Una URL de devolución de llamada indica a dónde se redirige al usuario después de 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

Su grupo de usuarios puede tener 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 de máquina a máquina (M2M), no puede conceder acceso a ninguno de sus 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. El cliente de aplicación puede escribir valores en atributos inmutables cuando se crea o se registra 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 la autenticación con nombre de usuario y contraseña, la OTP por correo electrónico y SMS, los autenticadores de clave de acceso, la autenticación personalizada con desencadenadores de Lambda y la actualización de tokens. Como práctica recomendada en materia de seguridad, utilice la autenticación SRP para la autenticación del nombre de usuario y la contraseña en las aplicaciones personalizadas.

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 Ámbitos, M2M y servidores de recursos.

URI de redireccionamiento predeterminado

Sustituye el `redirect_uri` parámetro en las solicitudes de autenticación de usuarios por otras de terceros IdPs. Configure esta configuración del cliente de la aplicación con el `DefaultRedirectURI` parámetro de una solicitud de UpdateUserPoolClientAPI CreateUserPoolCliento una solicitud. Esta URL también tiene que ser miembro de las

`CallbackURLs` del cliente de aplicación. Amazon Cognito redirige las sesiones autenticadas a esta URL cuando:

1. El cliente de aplicación tiene un proveedor de identidades asignado y varias URL de devolución de llamadas definidas. Su grupo de usuarios redirige las solicitudes de autenticación al servidor de autorización al URI de redireccionamiento predeterminado cuando las solicitudes no incluyen el parámetro `redirect_uri`.
2. El cliente de aplicación tiene un proveedor de identidades 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 el parámetro `redirect_uri` se redirigen a la única URL de devolución de llamada disponible.

Proveedores de identidades

Puede elegir algunos o todos los proveedores de identidad externos (IdPs) de su grupo de usuarios para autenticar a sus 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 en el inicio de sesión administrado. Puede asignar varios IdPs, pero debe asignar al menos uno. Para obtener más información sobre el uso de fuentes externas IdPs, consulteInicio de sesión en el grupo de usuarios con proveedores de identidad externos.

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 El punto de conexión 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 subvenciones 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 las comunicaciones de máquina a máquina (M2M). Authorization-code y las concesiones implícitas emiten fichas 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. Como alternativa, que no requiere un dominio de grupo de usuarios, un cliente de aplicaciones con un secreto de cliente y el flujo de `ALLOW_CLIENT_TOKEN_AUTH` autenticación puede obtener tokens de acceso M2M mediante la operación de la GetClientTokenAPI.

> **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.

Como los identificadores son una validación de los atributos del usuario, no son relevantes para la comunicación M2M y la concesión de credenciales de un cliente no los emite. Consulte Ámbitos, M2M y servidores de recursos.

La concesión de credenciales de cliente añade costes a tu AWS factura. Para obtener más información, consulte Precios de Amazon Cognito.

Creación de un cliente de aplicación

Consola de administración de AWS

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. Ambas opciones le piden que configure un cliente de aplicación con ajustes específicos de la aplicación.
4. Elija un tipo de aplicación que refleje la arquitectura de la aplicación.

   Important

   La selección del tipo de aplicación también establece los valores predeterminados para la configuración de otros clientes de aplicaciones, incluida la posibilidad de que Amazon Cognito genere un secreto de cliente. La aplicación web y los tipos de Machine-to-machine aplicaciones tradicionales generan un secreto de cliente. Para obtener más información, consulte Tipos de cliente de aplicación.

5. Asigne un nombre a la aplicación con un identificador fácil de entender.
6. Introduzca una URL de retorno.
7. Elija Create app client (Crear cliente de aplicación). Puede cambiar las opciones avanzadas después de crear el cliente de aplicación.
8. Amazon Cognito le devuelve a los detalles del cliente de aplicación. Para acceder al código de ejemplo de su aplicación, seleccione una plataforma en la pestaña Guía de configuración rápida.

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 de API. CreateUserPoolClient Debe especificar un valor para todos los parámetros que no desee establecer en un valor predeterminado.

Actualizar un cliente de aplicación para un grupo de usuarios (AWS CLI and 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 cliente de aplicaciones para grupos de usuarios (AWS CLI and 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 and 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 cliente de aplicación de grupo de usuarios (AWS CLI and 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

Implementación de flujos de autenticación

Ya sea que esté implementando un inicio de sesión administrado o una interfaz de aplicación personalizada con un AWS SDK para la autenticación, debe configurar el cliente de la aplicación para los tipos de autenticación que desee implementar. La siguiente información describe la configuración de los flujos de autenticación en los clientes de aplicación y en la aplicación.

App client supported flows

Puede configurar los flujos compatibles para los clientes de su aplicación en la consola de Amazon Cognito o con la API de un AWS SDK. Después de configurar el cliente de la aplicación para que admita estos flujos, puede implementarlos en la aplicación.

El siguiente procedimiento configura los flujos de autenticación disponibles para un cliente de aplicación con la consola Amazon Cognito.

Cómo configurar un cliente de aplicación para flujos de autenticación (consola)

1. Inicie sesión en la consola AWS de grupos de usuarios de Amazon Cognito y navegue hasta ella. Elija un grupo de usuarios o cree uno nuevo.
2. En la configuración de su grupo de usuarios, seleccione el menú Clientes de aplicación. Elija un cliente de aplicación o cree uno nuevo.
3. En Información del cliente de aplicación, seleccione Editar.
4. En Flujos de clientes de aplicación, elija los flujos de autenticación que desee admitir.

Para configurar un cliente de aplicaciones para los flujos de autenticación () API/SDK

Para configurar los flujos de autenticación disponibles para un cliente de aplicaciones con la API de Amazon Cognito, defina el valor de `ExplicitAuthFlows` en una solicitud

CreateUserPoolClientto UpdateUserPoolClient. A continuación, puede ver un ejemplo donde se proporciona a un cliente una contraseña remota segura (SRP) y una autenticación basada en opciones.

```
"ExplicitAuthFlows": [
   "ALLOW_USER_AUTH",
   "ALLOW_USER_SRP_AUTH"
]
```

Al configurar los flujos compatibles con el cliente de aplicación, puede especificar las opciones y valores de API siguientes.

Soporte de flujo del cliente de aplicación

Flujo de autenticación	Compatibilidad	Consola	API
Choice-basedautenticación	Server-side, del lado del cliente	Seleccionar un tipo de autenticación al iniciar sesión	`ALLOW_USER_AUTH`
Sign-in con contraseñas persistentes	Client-side	Inicio de sesión con nombre de usuario y contraseña	`ALLOW_USER_PASSWORD_AUTH`
Sign-in con contraseñas persistentes y carga segura	Server-side, del lado del cliente	Inicio de sesión con una contraseña remota segura (SRP)	`ALLOW_USER_SRP_AUTH`
Actualiza los tokens	Server-side, del lado del cliente	Obtener nuevos tokens de usuario de las sesiones autenticadas existentes	`ALLOW_REFRESH_TOKEN_AUTH`
Server-side autenticación	Server-side	Inicio de sesión con credenciales	`ALLOW_ADMIN_USER_PASSWORD_AUTH`

Flujo de autentica ción	Compatibilidad	Consola	API
		administrativas en el servidor	
Autenticación personalizada	Server-side y aplicaciones personalizadas del lado del cliente. No es compatible con el inicio de sesión administrado.	Iniciar sesión con flujos de autentica ción personalizados desde desencade nadores de Lambda	`ALLOW_CUS TOM_AUTH`

Implement flows in your application

El inicio de sesión gestionado hace que las opciones de autenticación configuradas estén disponibles automáticamente en las páginas de inicio de sesión. En las aplicaciones personalizadas, inicie la autenticación con una declaración del flujo inicial.

- Para elegir entre una lista de opciones de flujo para un usuario, declare la autenticación basada en opciones con el flujo `USER_AUTH`. Este flujo tiene métodos de autenticación que no están disponibles en los flujos de autenticación basados en el cliente, como la autenticación con clave de acceso y sin contraseña.
- Para elegir su flujo de autenticación por adelantado, declare la autenticación basada en el cliente con cualquier otro flujo que esté disponible en su cliente de aplicación.

Al iniciar sesión como usuario, el cuerpo de la AdminInitiateAuthsolicitud InitiateAuthdebe incluir un `AuthFlow` parámetro.

Choice-based autenticación:

```
"AuthFlow": "USER_AUTH"
```

Client-based autenticación con SRP:

```
"AuthFlow": "USER_SRP_AUTH"
```

Factores que debe tener en cuenta sobre la autenticación con grupos de usuarios

Tenga en cuenta la siguiente información al diseñar el modelo de autenticación con grupos de usuarios de Amazon Cognito.

Los flujos de autenticación en el inicio de sesión administrado y en la interfaz de usuario alojada

El inicio de sesión administrado tiene más opciones de autenticación que la interfaz de usuario alojada clásica. Por ejemplo, los usuarios solo pueden realizar la autenticación sin contraseña y con clave de acceso en el inicio de sesión administrado.

Los flujos de autenticación personalizados solo están disponibles en la autenticación AWS del SDK

No puede realizar flujos de autenticación personalizados ni autenticaciones personalizadas con activadores Lambda con el inicio de sesión administrado ni con la interfaz de usuario alojada clásica. La autenticación personalizada está disponible en la autenticación con AWS SDK.

Inicio de sesión administrado para un proveedor de identidades (IdP) externo

No puedes iniciar sesión a los usuarios a través de terceros IdPs en la autenticación con AWS los SDK. Debe implementar el inicio de sesión gestionado o la clásica interfaz de usuario alojada, redirigir al objeto de autenticación resultante y IdPs, a continuación, procesarlo con las bibliotecas OIDC de la aplicación. Para obtener más información sobre el inicio de sesión administrado, consulte Inicio de sesión administrado de grupos de usuarios.

Efecto de la autenticación sin contraseña en otras características de usuario

La activación del inicio de sesión sin contraseña con contraseñas de un solo uso o claves de acceso en el grupo de usuarios y en el cliente de aplicación repercute en la creación y migración de los usuarios. Cuando el inicio de sesión sin contraseña está activo, ocurre lo siguiente:

1. Los administradores pueden crear usuarios sin contraseñas. La plantilla de mensaje de invitación predeterminada cambia y ya no incluye el marcador de posición de contraseñas `{###}`. Para obtener más información, consulte Creación de cuentas de usuario como administrador.
2. En el SDK-based SignUpcaso de las operaciones, los usuarios no están obligados a proporcionar una contraseña al registrarse. El inicio de sesión administrado y la interfaz de usuario alojada requieren una contraseña en la página de registro, aunque la autenticación sin contraseña esté permitida. Para obtener más información, consulte Inscripción y confirmación de cuentas de usuario.

3. Los usuarios importados de un archivo CSV pueden iniciar sesión inmediatamente con opciones sin contraseña, sin necesidad de restablecer la contraseña, si sus atributos incluyen una dirección de correo electrónico o un número de teléfono para una opción de inicio de sesión sin contraseña disponible. Para obtener más información, consulte Importación de usuarios en grupos de usuarios desde un archivo CSV.
4. La autenticación sin contraseña no invoca el activador Lambda de migración de usuarios.
5. Los usuarios que inician sesión con una contraseña de un solo uso (OTP) como primer factor no pueden añadir un factor de autenticación multifactor (MFA) a su sesión. Las claves de paso con verificación de usuario pueden cumplir los requisitos de MFA cuando se configuran con. `MULTI_FACTOR_WITH_USER_VERIFICATION`

Las URL de los actores de confianza de las clave de acceso no pueden figurar en la lista pública de sufijos

Puede usar nombres de dominio de su propiedad, como `www.example.com`, como ID del actor de confianza (RP) en la configuración de su clave de acceso. Esta configuración está pensada para admitir aplicaciones personalizadas que se ejecutan en dominios de su propiedad. La lista pública de sufijos, o PSL, contiene dominios globales protegidos. Amazon Cognito devuelve un error cuando intenta establecer la URL de un RP en un dominio de la PSL.

Duración del flujo de la sesión de autenticación

En función de las características de su grupo de usuarios, puede responder a varios desafíos `RespondToAuthChallenge` antes de que su aplicación `InitiateAuth` 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. El inicio de sesión administrado establece la duración de la sesión en 3 minutos para la autenticación

multifactorial y 8 minutos para el cambio forzoso de la contraseña durante el primer inicio de sesión del usuario final con una contraseña temporal.

Amazon Cognito console

Para configurar la duración de la sesión del flujo de autenticación del cliente de la aplicación (Consola de administración de AWS)

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 Duración de la sesión del flujo de autenticación a la duración de validez que desee, en minutos, para los códigos MFA de SMS y correo electrónico. Esto también cambia la cantidad de tiempo que tiene cualquier usuario para completar cualquier desafío de autenticación en el cliente de la aplicación.
4. Seleccione Save changes (Guardar cambios).

User pools 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 Application-specific configuración con clientes de aplicaciones.

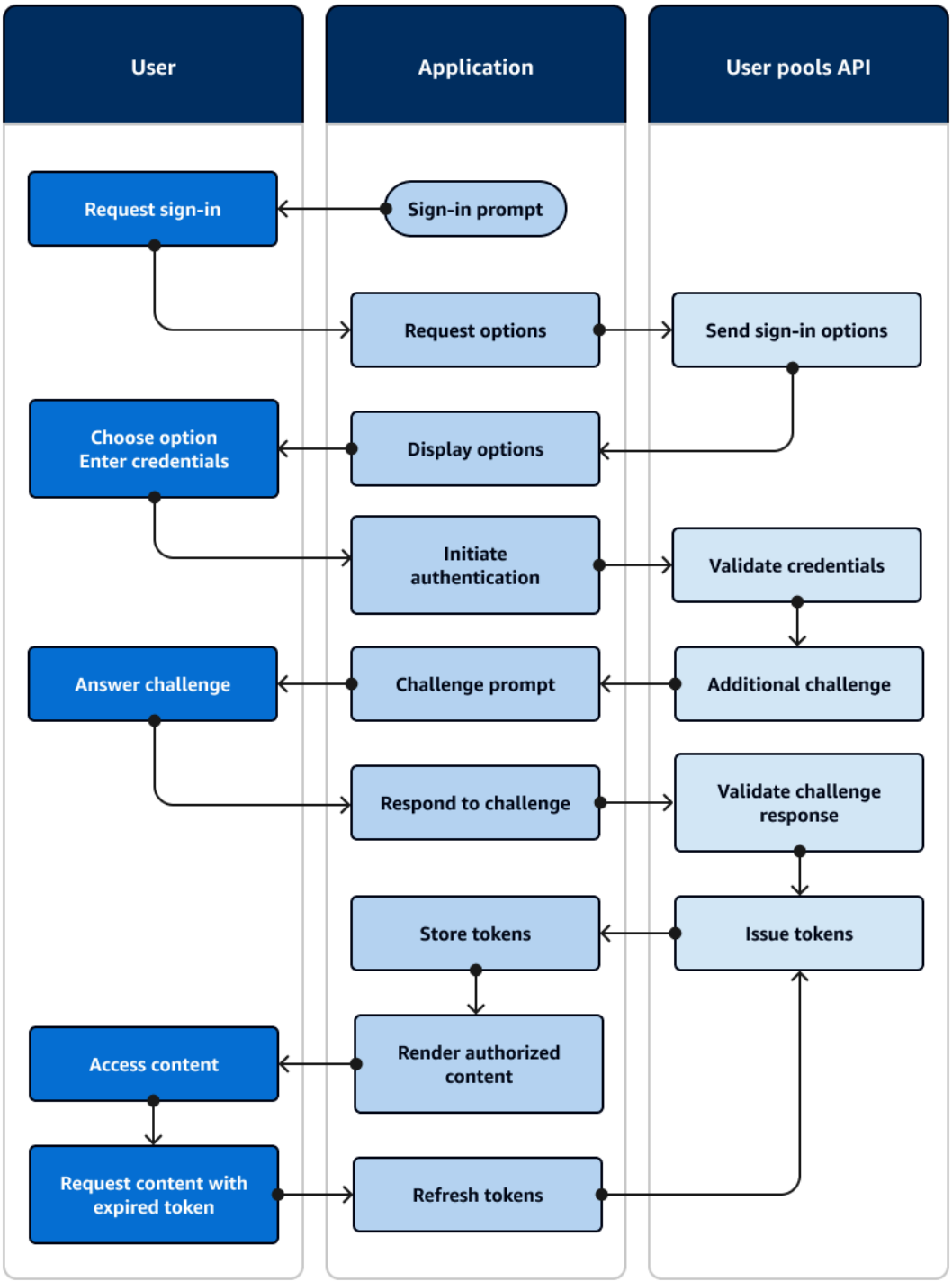
Comportamiento de bloqueo por intentos de inicio de sesión con error

Tras cinco intentos fallidos de inicio de sesión con la contraseña de un usuario, independientemente de si se solicitan con operaciones no autenticadas o de IAM-authorized API, 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.

Un ejemplo de sesión de autenticación

El siguiente diagrama y la guía paso a paso ilustran un escenario típico en el que un usuario inicia sesión en una aplicación. La aplicación de ejemplo presenta al usuario varias opciones de inicio de sesión. Para seleccionar una opción, el usuario debe introducir sus credenciales, proporcionar un factor de autenticación adicional e iniciar sesión.



Imagine una aplicación con una página de inicio de sesión en la que los usuarios puedan iniciar sesión con un nombre de usuario y una contraseña, solicitar un código de un solo uso en un mensaje de correo electrónico o elegir una opción de huella digital.

1. Sign-in aviso: la aplicación muestra una pantalla de inicio con un botón de inicio de sesión.
2. Solicitar inicio de sesión: el usuario selecciona Iniciar sesión. Desde una cookie o desde la memoria caché, la aplicación recupera su nombre de usuario o le pide que lo introduzca.
3. Opciones de solicitud: su aplicación solicita las opciones de inicio de sesión del usuario mediante una solicitud de API `InitiateAuth` con el flujo `USER_AUTH`, en la que se solicitan los métodos de inicio de sesión disponibles para el usuario.
4. Envío de opciones de inicio de sesión: Amazon Cognito responde con `PASSWORD`, `EMAIL_OTP` y `WEB_AUTHN`. La respuesta incluye un identificador de sesión para que lo reproduzca en la siguiente respuesta.
5. Opciones de visualización: su aplicación muestra los elementos de la interfaz de usuario para que el usuario introduzca su nombre de usuario y contraseña, obtenga un código de un solo uso o escanee su huella digital.
6. Elija option/Enter las credenciales: el usuario introduce su nombre de usuario y contraseña.
7. Inicio de la autenticación: su aplicación proporciona la información de inicio de sesión del usuario con una solicitud de API `RespondToAuthChallenge` que confirma el inicio de sesión con nombre de usuario y contraseña y proporciona el nombre de usuario y la contraseña.
8. Validación de las credenciales: Amazon Cognito confirma las credenciales del usuario.
9. Desafío adicional: el usuario tiene la autenticación multifactor configurada con una aplicación de autenticación. Amazon Cognito devuelve un desafío `SOFTWARE_TOKEN_MFA`.
10. Petición del desafío: la aplicación muestra un formulario en el que se solicita una contraseña temporal de un solo uso (TOTP) desde la aplicación de autenticación del usuario.
11. Respuesta al desafío: el usuario envía la TOTP.
12. Responder al desafío: en otra solicitud `RespondToAuthChallenge`, su aplicación proporciona la TOTP del usuario.
13. Validación de la respuesta al desafío: Amazon Cognito confirma el código del usuario y determina que su grupo de usuarios está configurado para no plantear desafíos adicionales al usuario actual.
14. Emisión de tokens: Amazon Cognito devuelve los tokens web JSON (JWT) de ID, acceso y actualización. La autenticación inicial del usuario está completa.
15. Almacenamiento de los tokens: su aplicación almacena en caché los tokens del usuario para poder hacer referencia a los datos del usuario, autorizar el acceso a los recursos y actualizar los tokens cuando caduquen.
16. Representación de contenido autorizado: su aplicación determina el acceso del usuario a los recursos en función de su identidad y roles, y entrega el contenido de la aplicación.

17. Acceso al contenido: el usuario ha iniciado sesión y comienza a usar la aplicación.
18. Solicitud de contenido con un token caducado: posteriormente, el usuario solicita un recurso que requiere autorización. El token en caché del usuario ha caducado.
19. Tokens de actualización: su aplicación hace una solicitud `InitiateAuth` con el token de actualización guardado por el usuario.
20. Emisión de tokens: Amazon Cognito devuelve un nuevo ID y accede a JWT. La sesión del usuario se actualiza de forma segura sin solicitar credenciales adicionales.

Puede usar activadores AWS Lambda 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 usar el flujo de autenticación del administrador para proteger los servidores back-end. Puede utilizar el flujo de autenticación de la migración de usuarios para posibilitar la migración de los usuarios sin necesidad de que estos restablezcan sus contraseñas.

Configuración de los métodos de autenticación para el inicio de sesión administrado

Puede invocar páginas de inicio de sesión administrado, un frontend web para la autenticación de grupos de usuarios, cuando desee que los usuarios inicien sesión, cierren sesión o restablezcan su contraseña. En este modelo, la aplicación importa las bibliotecas OIDC para procesar los intentos de autenticación basados en navegador con páginas de inicio de sesión administrado por grupos de usuarios. Las formas de autenticación disponibles para los usuarios dependen de la configuración del grupo de usuarios y del cliente de aplicación. Implemente el flujo `ALLOW_USER_AUTH` en su cliente de aplicación: Amazon Cognito solicitará a los usuarios que seleccionen un método de inicio de sesión entre las opciones disponibles. Implemente `ALLOW_USER_PASSWORD_AUTH` y asigne un proveedor de SAML: sus páginas de inicio de sesión indicarán a los usuarios la opción de introducir su nombre de usuario y contraseña o de conectarse con su IdP.

La consola de grupos de usuarios de Amazon Cognito puede ayudarle a empezar a configurar la autenticación de inicio de sesión administrado para su aplicación. Cuando cree un nuevo grupo de usuarios, especifique la plataforma para la que está desarrollando: la consola le proporcionará ejemplos de implementación de bibliotecas OIDC y OAuth con un código de inicio para implementar los flujos de inicio y cierre de sesión. Puede crear un inicio de sesión administrado con muchas implementaciones de actores de confianza del OIDC. Siempre que sea posible, le recomendamos que utilice bibliotecas de actores de confianza de OIDC certificadas. Para obtener más información, consulte Creación de una nueva aplicación en la consola de Amazon Cognito.

Por lo general, las bibliotecas de las partes dependientes del OIDC comprueban periódicamente el `.well-known/openid-configuration` punto final del grupo de usuarios para determinar las direcciones URL del emisor, como el punto final del token y el punto final de autorización. Como práctica recomendada, implemente este comportamiento de detección automática cuando tenga la opción de hacerlo. La configuración manual de los puntos de conexión del emisor introduce la posibilidad de que se produzcan errores. Por ejemplo, podría cambiar el dominio del grupo de usuarios. La ruta a `openid-configuration` no está vinculada al dominio del grupo de usuarios, por lo que las aplicaciones que detectan automáticamente los puntos de enlace del servicio detectan automáticamente el cambio de dominio.

Configuración del grupo de usuarios para el inicio de sesión administrado

Es posible que desee permitir el inicio de sesión con varios proveedores para su aplicación, o quizá quiera usar Amazon Cognito como un directorio de usuarios independiente. Es posible que también desee recopilar los atributos de los usuarios, configurar y solicitar el MFA o solicitar direcciones de correo electrónico como nombres de usuario. No puede editar directamente los campos en el inicio de sesión administrado ni en la interfaz de usuario alojada. En lugar de eso, la configuración del grupo de usuarios establece automáticamente la gestión de los flujos de autenticación de los inicios de sesión administrados.

Los siguientes elementos de configuración del grupo de usuarios determinan los métodos de autenticación que Amazon Cognito presenta a los usuarios en el inicio de sesión administrado y en la interfaz de usuario alojada.

User pool options (Sign-in menu)

Las siguientes opciones se encuentran en el Sign-inmenú de un grupo de usuarios de la consola de Amazon Cognito.

Opciones de inicio de sesión para grupos de usuarios en Cognito

Tiene opciones para los nombres de usuario. Sus páginas de inicio de sesión administrado y de interfaz de usuario alojada solo aceptan nombres de usuario en los formatos que seleccione. Cuando, por ejemplo, configura un grupo de usuarios con el correo electrónico como única opción de inicio de sesión, sus páginas de inicio de sesión administrado solo aceptan nombres de usuario en formato de correo electrónico.

Atributos obligatorios.

Cuando establece un atributo como obligatorio en su grupo de usuarios, el inicio de sesión administrado solicita a los usuarios un valor para ese atributo cuando se registran.

Opciones para el inicio de sesión basado en opciones

Tiene la configuración para los métodos de autenticación en Choice-based authentication. Aquí puede activar o desactivar los métodos de autenticación, como el uso de la clave de acceso y el método sin contraseña. Estos métodos solo están disponibles para grupos de usuarios con dominios de inicio de sesión administrados y planes de características superiores al nivel Lite.

Multi-factor authentication

El inicio de sesión administrado y la interfaz de usuario alojada gestionan las operaciones de registro y autenticación de la MFA. Cuando se requiere MFA en su grupo de usuarios, sus páginas de inicio de sesión solicitan automáticamente a los usuarios que configuren su factor adicional. También piden a los usuarios que tienen una configuración de MFA que completen la autenticación con un código MFA. Cuando la MFA está desactivada o es opcional en su grupo de usuarios, las páginas de inicio de sesión no solicitan configurar la MFA.

Recuperación de cuentas de usuario

La configuración de autoservicio para la recuperación de cuentas de su grupo de usuarios determina si las páginas de inicio de sesión muestran un enlace en el que los usuarios puedan restablecer su contraseña.

User pool options (Domain menu)

Las siguientes opciones se encuentran en el menú Dominio de un grupo de usuarios de la consola de Amazon Cognito.

Dominio

La elección del dominio del grupo de usuarios establece la ruta del enlace que los usuarios abren cuando invoca sus navegadores para autenticarse.

Versión de marca

La versión de marca que elija determinará si el dominio del grupo de usuarios mostrará el inicio de sesión administrado o la interfaz de usuario alojada.

User pool options (Social and external providers menu)

La siguiente opción se encuentra en el menú Proveedores sociales y externos de un grupo de usuarios en la consola de Amazon Cognito.

Proveedores

Los proveedores de identidad (IdPs) que añada a su grupo de usuarios pueden permanecer activos o inactivos para cada cliente de aplicaciones del grupo de usuarios.

App client options

Las siguientes opciones se encuentran en el menú Clientes de aplicación de un grupo de usuarios de la consola de Amazon Cognito. Para revisar estas opciones, seleccione un cliente de aplicación de la lista.

Guía de configuración rápida

La guía de configuración rápida incluye ejemplos de código para una variedad de entornos de desarrolladores. Contienen las bibliotecas necesarias para integrar la autenticación de inicio de sesión administrado en su aplicación.

Información sobre el cliente de aplicación

Edite esta configuración IdPs para asignarla a la aplicación que representa el cliente de la aplicación actual. En las páginas de inicio de sesión administrado, Amazon Cognito muestra las opciones para los usuarios. Estas opciones se determinan a partir de los métodos asignados y el IdP. Por ejemplo, si asigna un IdP de SAML 2.0 con el nombre `MySAML` y un inicio de sesión para un grupo de usuarios local, sus páginas de inicio de sesión administrado mostrarán las indicaciones del método de autenticación y un botón para `MySAML`.

Configuración de autenticación

Edite esta configuración para establecer los métodos de autenticación de su aplicación. En las páginas de inicio de sesión administrado, Amazon Cognito muestra las opciones para los usuarios. Estas opciones se determinan a partir de la disponibilidad del grupo de usuarios como IdP y de los métodos que usted asigne. Por ejemplo, si asigna una autenticación `ALLOW_USER_AUTH` basada en opciones, las páginas de inicio de sesión administrado mostrarán las opciones disponibles, como introducir una dirección de correo electrónico e iniciar sesión con una clave de acceso. Las páginas de inicio de sesión administradas también muestran los botones de las páginas asignadas IdPs.

Páginas de inicio de sesión

Defina el efecto visual de sus páginas de inicio de sesión administrado o de las páginas interactivas para el usuario de la interfaz de usuario alojada con las opciones disponibles en

esta pestaña. Para obtener más información, consulte Aplicación de la creación de marca en las páginas de inicio de sesión administrado.

Administre los métodos de autenticación en AWS SDK

Los usuarios de los grupos de usuarios de Amazon Cognito pueden iniciar sesión con una variedad de opciones o factores de inicio de sesión inicial. En algunos casos, los usuarios pueden hacer un seguimiento con autenticación multifactor (MFA). Estos primeros factores incluyen el nombre de usuario y la contraseña, la contraseña de un solo uso, la clave de acceso y la autenticación personalizada. Para obtener más información, consulte Flujos de autenticación. Cuando su aplicación tiene componentes de interfaz de usuario integrados e importa un módulo AWS del SDK, debe crear la lógica de la aplicación para la autenticación. Debe elegir uno de los dos métodos principales y, a partir de ese método, los mecanismos de autenticación que desee implementar.

Puede implementar la autenticación basada en el cliente, en la que la aplicación, o el cliente, declare el tipo de autenticación por adelantado. La otra opción es la autenticación basada en opciones, en la que la aplicación recopila un nombre de usuario y solicita los tipos de autenticación disponibles para los usuarios. Puede implementar estos modelos juntos en la misma aplicación o dividirlos entre los clientes de la aplicación, según lo que necesite. Cada método tiene características que son únicas; por ejemplo, la autenticación personalizada, en el método basado en el cliente, y la autenticación sin contraseña, en el caso del método basado en opciones.

En las aplicaciones personalizadas que se autentican con la implementación del AWS SDK de la API de grupos de usuarios, debes estructurar las solicitudes de API para adaptarlas a la configuración del grupo de usuarios, la configuración del cliente de la aplicación y las preferencias del lado del cliente. Una sesión `InitiateAuth` que comience con un `AuthFlow` de `USER_AUTH` empieza con la autenticación basada en opciones. Amazon Cognito responde a su API con un desafío del método de autenticación preferido o una lista de opciones. Una sesión que comienza con `AuthFlow` de `CUSTOM_AUTH` pasa directamente a la autenticación personalizada con activadores Lambda.

Algunos métodos de autenticación están fijos en uno de los dos tipos de flujo y algunos métodos están disponibles en ambos.

Temas

- Choice-based authentication
- Client-based authentication
- Step-up autenticación con el flujo USER_AUTH

Choice-based authentication

La aplicación puede solicitar los siguientes métodos de autenticación en la autenticación basada en opciones. Declare estas opciones en el `PREFERRED_CHALLENGE` parámetro InitiateAutho o AdminInitiateAuthen el `ChallengeName` parámetro o. RespondToAuthChallengeAdminRespondToAuthChallenge

> **Important**
>
> Choice-based el inicio de sesión tiene dos requisitos previos. El cliente de la aplicación debe activar el flujo de **`ALLOW_USER_AUTH`** autenticación, y los métodos sin contraseña y con clave de paso (**`EMAIL_OTP`**,**`SMS_OTP`**,**`WEB_AUTHN`**) requieren un grupo de usuarios en el plan de funciones Essentials o Plus; no están disponibles en el nivel Lite. `PASSWORD`(incluido el SRP) está disponible en todos los planes de funciones. Si tus usuarios no ven un método basado en una elección, confirma tanto el flujo entre la aplicación y el cliente como el plan de funciones antes de continuar con la solución de problemas.

1. `EMAIL_OTP` y `SMS_OTP`

   Inicio de sesión sin contraseña con contraseñas de un solo uso

2. `WEB_AUTHN`

   Inicio de sesión sin contraseña con claves de acceso WebAuthn

3. `PASSWORD`

   Sign-in con contraseñas persistentes

   Sign-in con contraseñas persistentes y una carga segura

   MFA después del inicio de sesión

Para revisar estas opciones en su contexto de API, consulte. `ChallengeName` RespondToAuthChallenge

Choice-based el inicio de sesión genera un desafío en respuesta a tu solicitud inicial. Este desafío verifica que la opción solicitada esté disponible o proporciona una lista de las opciones disponibles. La aplicación puede mostrar estas opciones a los usuarios, quienes, a continuación, introducen las

credenciales del método de inicio de sesión preferido y proceden a autenticarse en las respuestas a las preguntas.

Dispone de las siguientes opciones basadas en opciones en su flujo de autenticación. Todas las solicitudes de este tipo requieren que su aplicación primero recopile un nombre de usuario o lo recupere de una memoria caché.

1. Solicite opciones solo con `AuthParameters` de `USERNAME`. Amazon Cognito devuelve un desafío `SELECT_CHALLENGE`. A partir de ahí, la aplicación puede solicitar al usuario que seleccione un desafío y devuelva esta respuesta a su grupo de usuarios.
2. Solicite un desafío preferido con `AuthParameters` de `PREFERRED_CHALLENGE` y los parámetros del desafío que prefiera, si los hubiera. Por ejemplo, si solicita un `PREFERRED_CHALLENGE` de `PASSWORD_SRP`, también debe incluir `SRP_A`. Si el usuario, el grupo de usuarios y el cliente de la aplicación están configurados para el desafío preferido, Amazon Cognito responde con el siguiente paso de ese desafío, por ejemplo, `PASSWORD_VERIFIER` en el `PASSWORD_SRP` flujo o CodeDeliveryDetailsen los flujos `EMAIL_OTP` and`SMS_OTP`. Si el desafío preferido no está disponible, Amazon Cognito responde con `SELECT_CHALLENGE` y una lista de los desafíos disponibles.
3. Deje que los usuarios inicien sesión en primer lugar y, luego, solicite las opciones de autenticación basadas en opciones. Una GetUserAuthFactorssolicitud con el token de acceso de un usuario que ha iniciado sesión devuelve sus factores de autenticación basados en opciones disponibles y su configuración de MFA. Con esta opción, un usuario puede iniciar sesión primero con el nombre de usuario y la contraseña y, a continuación, activar otra forma de autenticación. También puede usar esta operación para seleccionar opciones adicionales para un usuario que haya iniciado sesión con un desafío preferido.

A fin de configurar el cliente de aplicación para la autenticación basada en opciones, añada `ALLOW_USER_AUTH` a los flujos de autenticación permitidos. También debe elegir los factores basados en opciones que desee permitir en la configuración de su grupo de usuarios. El siguiente proceso ilustra cómo elegir los factores de autenticación basados en opciones.

Amazon Cognito console

Cómo configurar las opciones de autenticación basadas en opciones en un grupo de usuarios

1. Inicie sesión en la consola AWS de grupos de usuarios de Amazon Cognito y navegue hasta ella. Elija un grupo de usuarios o cree uno nuevo.

2. En la configuración de su grupo de usuarios, seleccione el Sign-inmenú. Busque Opciones para el inicio de sesión basado en opciones y seleccione Editar.
3. La opción Contraseña siempre está disponible. Esto incluye los flujos `PASSWORD` y `PASSWORD_SRP`. Seleccione las opciones adicionales que desee añadir a las opciones de sus usuarios. Puede agregar una clave de acceso para `WEB_AUTHN`, una contraseña única en mensaje de correo electrónico para `EMAIL_OTP` y una contraseña única en mensaje de SMS para `SMS_OTP`.
4. Seleccione Save changes (Guardar cambios).

API/SDK

El siguiente cuerpo parcial CreateUserPoolo de UpdateUserPoolsolicitud configura todas las opciones disponibles para la autenticación basada en elecciones.

```
"Policies": {
    "SignInPolicy": {
        "AllowedFirstAuthFactors": [
            "PASSWORD",
            "WEB_AUTHN",
            "EMAIL_OTP",
            "SMS_OTP"
        ]
    }
},
```

Client-based authentication

Client-based la autenticación admite los siguientes flujos de autenticación. Declare estas opciones en el `AuthFlow` parámetro InitiateAutho AdminInitiateAuth.

1. `USER_PASSWORD_AUTH` y `ADMIN_USER_PASSWORD_AUTH`

   Sign-in con contraseñas persistentes

   MFA después del inicio de sesión

   Este flujo de autenticación es equivalente a `PASSWORD` en la autenticación basada en opciones.

2. `USER_SRP_AUTH`

Sign-in con contraseñas persistentes y una carga segura

MFA después del inicio de sesión

Este flujo de autenticación es equivalente a `PASSWORD_SRP` en la autenticación basada en opciones.

3. `REFRESH_TOKEN_AUTH`

   Tokens de actualización

   Este flujo de autenticación solo está disponible en la autenticación basada en el cliente.

4. `CUSTOM_AUTH`

   Autenticación personalizada

   Este flujo de autenticación solo está disponible en la autenticación basada en el cliente.

Con la autenticación basada en el cliente, Amazon Cognito da por sentado que usted ha determinado cómo quiere autenticarse el usuario antes de que este comience los flujos de autenticación. La lógica para determinar el factor de inicio de sesión que un usuario quiere proporcionar debe determinarse con la configuración predeterminada o con mensajes personalizados; luego, debe declararse en la primera solicitud a su grupo de usuarios. La solicitud `InitiateAuth` declara un `AuthFlow` de inicio de sesión que se corresponde directamente con una de las opciones de la lista, como `USER_SRP_AUTH`. Con esta declaración, la solicitud también incluye los parámetros para iniciar la autenticación, como `USERNAME`, `SECRET_HASH` y `SRP_A`. Amazon Cognito podría complementar esta solicitud con desafíos adicionales, como `PASSWORD_VERIFIER` para el SRP o `SOFTWARE_TOKEN_MFA` para el inicio de sesión con contraseña, con TOTP y MFA.

A fin de configurar el cliente de aplicación para la autenticación basada en el cliente, añada cualquier flujo de autenticación distinto de `ALLOW_USER_AUTH` en los flujos de autenticación permitidos. Algunos ejemplos son`ALLOW_USER_PASSWORD_AUTH`,`ALLOW_CUSTOM_AUTH`,`ALLOW_REFRESH_TOKEN_AUTH`. Para permitir los flujos de autenticación basados en el cliente, no se requiere ninguna configuración adicional del grupo de usuarios.

Step-up autenticación con el flujo USER_AUTH

Se solicita una autenticación progresiva en el `USER_AUTH` flujo existente con las operaciones y. InitiateAuthRespondToAuthChallenge No hay nuevas operaciones de API. Las operaciones correspondientes del lado del servidor admiten los mismos parámetros AdminInitiateAuthy AdminRespondToAuthChallengecomportamientos de escalonamiento y, opcionalmente, también `AdminInitiateAuth` pueden utilizar un token de acceso. Este flujo funciona tanto cuando un usuario tiene un token de identificación existente como cuando se autentica en un nivel objetivo por primera vez. Para obtener más información sobre el flujo, consulteUn ejemplo de sesión de autenticación.

Para obtener más información sobre los niveles de ACR y los valores de AMR, consulte. the section called “Niveles de autenticación con notificaciones de ACR y AMR”

> **Note**
>
> Step-up la autenticación requiere el plan de funciones Essentials o Plus.

Usted introduce los parámetros de escalonamiento en el `AuthParameters` mapa de su `InitiateAuth` (o`AdminInitiateAuth`) solicitud: los nuevos `TARGET_ACR_VALUES` y`MAX_AGE`, junto con los existentes `USERNAME`, `ACCESS_TOKEN`, y`PREFERRED_CHALLENGE`. Para conocer la definición y las restricciones de cada parámetro, consulte InitiateAuthy AdminInitiateAuthen la referencia de la API de grupos de usuarios de Amazon Cognito.

En la siguiente tabla se describe cómo Amazon Cognito responde a las combinaciones de estos parámetros.

USER_AUTH intensifica el comportamiento de autenticación

Parameters	Comportamiento
`TARGET_ACR_VALUES` y, con satisfacción `ACCESS_TOKEN` `MAX_AGE`	Amazon Cognito hace que el usuario pase de los factores actuales del token al nivel satisfactorio de mayor prioridad .
`TARGET_ACR_VALUES` y`ACCESS_TOKEN` , sin estar satisfecho `MAX_AGE`	Amazon Cognito autentica al usuario desde cero hasta el nivel objetivo.

Parameters	Comportamiento
`TARGET_ACR_VALUES` sin `ACCESS_TOKEN`	Amazon Cognito autentica al usuario desde cero hasta el nivel objetivo.
Ninguno de los dos parámetros	El `USER_AUTH` flujo se comporta como lo hace sin una autenticación intensificada.

Tenga en cuenta el siguiente comportamiento al diseñar un flujo de autenticación escalonado.

- Step-up la autenticación es todo o nada. Si alcanzar el nivel objetivo requiere varios desafíos, por ejemplo, una contraseña y un TOTP para el nivel 4, y el usuario no supera alguno de los desafíos, el intento de mejora fracasa por completo. Amazon Cognito no emite tokens intermedios.
- Si el token presentado ya alcanza o supera el nivel solicitado, Amazon Cognito devuelve un error en lugar de emitir un nuevo token. Esto impide que las aplicaciones utilicen la autenticación escalonada como mecanismo de actualización de los tokens.
- El usuario ya debe tener los factores necesarios para satisfacer un nivel solicitado. Si el usuario no tiene ningún factor configurado para cumplir el nivel objetivo solicitado, Amazon Cognito devuelve un error. Step-upla autenticación no solicita al usuario que configure nuevos factores.
- El MFA es un suelo. Si la configuración de MFA del usuario o del grupo de usuarios requiere MFA, el inicio de sesión inicial sigue exigiendo y solicitando ese desafío de MFA incluso cuando el nivel objetivo es inferior y, de otro modo, no requeriría MFA. El MFA que solicita la protección contra amenazas (funciones de seguridad avanzadas) también se aplica en el inicio de sesión inicial, incluso cuando el nivel de ACR objetivo solicitado no requeriría el MFA. `TARGET_ACR_VALUES`no se puede suprimir un mensaje de MFA obligatorio. Como máximo, el objetivo puede influir en cuál de los varios factores de MFA habilitados que solicita Amazon Cognito. Si una solicitud escalonada se dirige a un nivel inferior, por ejemplo, el nivel 1, para un usuario que debe usar MFA, Amazon Cognito aplica silenciosamente el MFA requerido. Incrementa el nivel de usuario y no devuelve ningún error.

Para solicitar una autenticación avanzada se `TARGET_ACR_VALUES` requiere el plan de funciones Essentials o Plus. Para obtener más información, consulte Requisitos del plan de funciones.

Flujos de autenticación

El proceso de autenticación con grupos de usuarios de Amazon Cognito puede describirse mejor como un flujo en el que los usuarios toman una decisión inicial, envían credenciales y responden a desafíos adicionales. Cuando implementa la autenticación de inicio de sesión administrado en su aplicación, Amazon Cognito gestiona el flujo de estas solicitudes y desafíos. Al implementar flujos con un AWS SDK en el back-end de la aplicación, debe crear la lógica de las solicitudes, solicitar a los usuarios que aporten información y responder a los desafíos.

Como administrador de aplicaciones, las características de su usuario, sus requisitos de seguridad y su modelo de autorización ayudan a determinar cómo desea permitir que los usuarios inicien sesión. Hágase las siguientes preguntas:

- ¿Quiero permitir que los usuarios inicien sesión con credenciales de otros proveedores de identidad (IdPs)?
- ¿Un nombre de usuario y una contraseña son suficiente como prueba de identidad?
- ¿Podrían interceptarse mis solicitudes de autenticación de nombre de usuario y contraseña? ¿Deseo que mi aplicación transmita contraseñas o que negocie la autenticación mediante hashes y sales?
- ¿Deseo permitir que los usuarios se salten la introducción de contraseñas y reciban una contraseña de un solo uso que les permita iniciar sesión?
- ¿Quiero permitir que los usuarios inicien sesión con una huella digital, el rostro o una clave de seguridad de hardware?
- ¿Cuándo quiero pedir la autenticación multifactor (MFA) si es que querré hacerlo?
- ¿Deseo que las sesiones de usuario se mantengan sin tener que volver a solicitar las credenciales?
- ¿Deseo ampliar mi modelo de autorización más allá de las funciones integradas de Amazon Cognito?

Cuando tenga las respuestas a estas preguntas, podrá ver cómo activar las características pertinentes e implementarlas en las solicitudes de autenticación que realice su aplicación.

Después de configurar los flujos de inicio de sesión para un usuario, puedes comprobar su estado actual para ver si tiene MFA y factores de autenticación basados en elecciones con las solicitudes a la operación de la API. GetUserAuthFactors Esta operación requiere la autorización con el token de

acceso de un usuario que haya iniciado sesión. Devuelve los factores de autenticación del usuario y la configuración de MFA.

Temas

- Sign-in con un tercero IdPs
- Sign-in con contraseñas persistentes
- Sign-in con contraseñas persistentes y una carga segura
- Inicio de sesión sin contraseña con contraseñas de un solo uso
- Inicio de sesión sin contraseña con claves de acceso WebAuthn
- MFA después del inicio de sesión
- Tokens de actualización
- Autenticación personalizada
- Flujo de autenticación de migración de usuarios

Sign-in con un tercero IdPs

Los grupos de usuarios de Amazon Cognito actúan como intermediario de las sesiones de autenticación entre servicios IdPs como Sign in with Apple, Login with Amazon y OpenID Connect (OIDC). Este proceso también se denomina inicio de sesión federado o autenticación federada. La autenticación federada no utiliza ninguno de los flujos de autenticación que puede crear en el cliente de su aplicación. En su lugar, usted asigna el grupo de usuarios configurado al cliente de IdPs su aplicación. El inicio de sesión federado ocurre cuando los usuarios seleccionan su IdP en el inicio de sesión administrado o cuando la aplicación invoca una sesión con una redirección a su página de inicio de sesión de IdP.

Con el inicio de sesión federado, delega los factores de autenticación principales y de MFA al IdP del usuario. Amazon Cognito no añade los demás flujos avanzados de esta sección a un usuario federado a menos que los vincule a un usuario local. Los usuarios federados no vinculados tienen nombres de usuario, pero son un almacén de datos de atributos mapeados que normalmente no se utilizan para el inicio de sesión independiente del flujo con navegador.

Recursos de implementación

- Inicio de sesión en el grupo de usuarios con proveedores de identidad externos

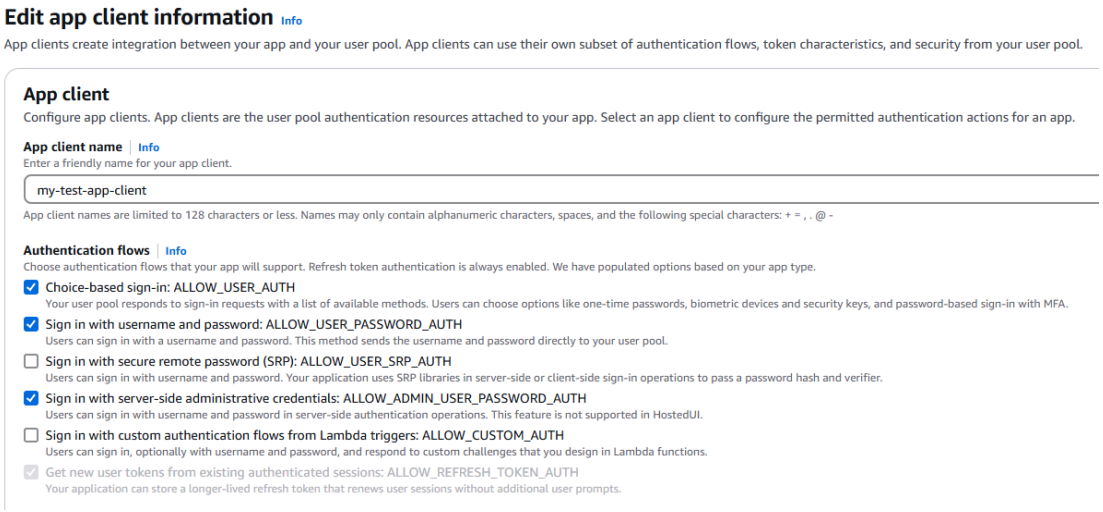
Sign-in con contraseñas persistentes

En los grupos de usuarios de Amazon Cognito, cada usuario tiene un nombre de usuario. Podría ser un número de teléfono, una dirección de correo electrónico o un identificador elegido o proporcionado por el administrador. Los usuarios de este tipo pueden iniciar sesión con su nombre de usuario y contraseña y, de forma opcional, proporcionar una MFA. Los grupos de usuarios pueden iniciar sesión con nombre de usuario y contraseña con operaciones públicas o de IAM-authorized API y métodos del SDK. La aplicación puede enviar directamente la contraseña a su grupo de usuarios para su autenticación. Su grupo de usuarios responde con desafíos adicionales o con los tokens web JSON (JWT) que son el resultado de una autenticación correcta.

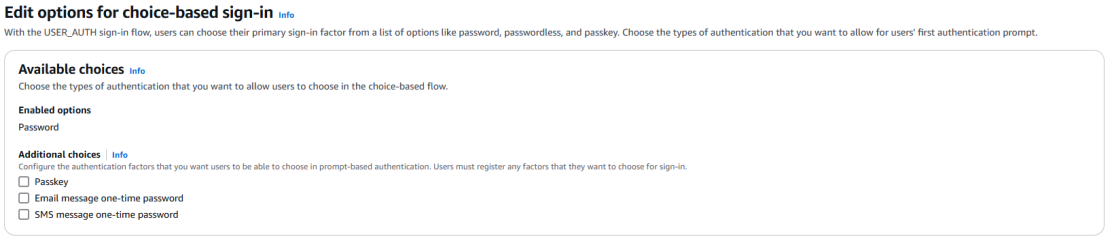
Activate password sign-in

Para activar la autenticación basada en el cliente con nombre de usuario y contraseña, configure el cliente de la aplicación para que lo permita. En la consola de Amazon Cognito, navegue hasta el menú de clientes de aplicaciones en la configuración de su grupo de usuarios. Para permitir el inicio de sesión con una contraseña simple en una aplicación móvil o nativa en el cliente, edite un cliente de aplicación y seleccione Iniciar sesión con nombre de usuario y contraseña: ALLOW_USER_PASSWORD_AUTH en Flujos de autenticación. Para permitir el inicio de sesión con una contraseña simple en una aplicación del servidor, edite el cliente de aplicación y elija Iniciar sesión con las credenciales administrativas del servidor: ALLOW_ADMIN_USER_PASSWORD_AUTH.

Para activar la autenticación basada en opciones con nombre de usuario y contraseña, configure el cliente de aplicación para que lo permita. Edite el cliente de la aplicación y elija Choice-based iniciar sesión: ALLOW_USER_AUTH.



Para comprobar que la autenticación mediante contraseña está disponible en los flujos de autenticación basados en elecciones, navega hasta el Sign-in menú y consulta la sección de Opciones para el inicio de sesión basado en elecciones. Puede iniciar sesión con una autenticación con contraseña simple si la contraseña está visible en Opciones disponibles. La opción de contraseña incluye las variantes de autenticación simple y SRP con nombre de usuario y contraseña.



Configure `ExplicitAuthFlows` con las opciones de autenticación de nombre de usuario y contraseña que prefiera en una solicitud o. CreateUserPoolClientUpdateUserPoolClient

```
"ExplicitAuthFlows": [
   "ALLOW_USER_PASSWORD_AUTH",
   "ALLOW_ADMIN_USER_PASSWORD_AUTH",
   "ALLOW_USER_AUTH"
]
```

En una UpdateUserPoolsolicitud CreateUserPoolo, configúrela `Policies` con los flujos de autenticación basados en opciones que desee admitir. El valor `PASSWORD` en `AllowedFirstAuthFactors` incluye las opciones de flujo de autenticación con contraseña simple y SRP.

```
"Policies": {
   "SignInPolicy": {
      "AllowedFirstAuthFactors": [
         "PASSWORD",
         "EMAIL_OTP",
         "WEB_AUTHN"
      ]
   }
}
```

Choice-based sign-in with a password

Para iniciar sesión en una aplicación con la autenticación de nombre de usuario y contraseña, configure el cuerpo de su AdminInitiateAuthsolicitud de la siguiente manera. InitiateAuth Esta solicitud de inicio de sesión se realiza correctamente o continúa hasta el siguiente desafío si el usuario actual cumple los requisitos para la autenticación con nombre de usuario y contraseña. De lo contrario, responde con una lista de los desafíos de autenticación de factor principal disponibles. Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PREFERRED_CHALLENGE" : "PASSWORD",
      "PASSWORD" : "[User's password]"
   },
   "ClientId": "1example23456789"
}
```

También puede omitir el valor de `PREFERRED_CHALLENGE` y recibir una respuesta que contenga una lista de los factores de inicio de sesión aptos para el usuario.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser"
   },
   "ClientId": "1example23456789"
}
```

Si no ha enviado un desafío preferido o el usuario enviado no reúne los requisitos para su desafío preferido, Amazon Cognito devolverá una lista de opciones en `AvailableChallenges`. Si `AvailableChallenges` incluye un `ChallengeName` de`PASSWORD`, puede continuar con la autenticación con una respuesta RespondToAuthChallengeo AdminRespondToAuthChallengeimpugnación en el siguiente formato. Debe pasar un parámetro `Session` que asocie la respuesta al desafío con la respuesta de la API a su solicitud de inicio de sesión inicial. Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

```
{
   "ChallengeName": "PASSWORD",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "PASSWORD" : "[User's Password]"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response"
}
```

Amazon Cognito responde a las solicitudes de desafío preferido aptas y satisfactorias y a las respuestas a desafíos PASSWORD con tokens o un desafío adicional obligatorio, como la autenticación multifactor (MFA).

Client-based sign-in with a password

Para iniciar sesión en una aplicación del cliente con la autenticación de nombre de usuario y contraseña, configura el cuerpo de la solicitud de la siguiente manera. InitiateAuth Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

```
{
   "AuthFlow": "USER_PASSWORD_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PASSWORD" : "[User's password]"
   },
   "ClientId": "1example23456789"
}
```

Para iniciar sesión en una aplicación del servidor con autenticación de nombre de usuario y contraseña, configura el cuerpo de la solicitud de la siguiente manera. AdminInitiateAuth La aplicación debe firmar esta solicitud con credenciales. AWS Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

```
{
   "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PASSWORD" : "[User's password]"
   },
```

```
    "ClientId": "1example23456789"
}
```

Amazon Cognito responde a las solicitudes exitosas con tokens o un desafío adicional requerido, como la autenticación multifactor (MFA).

Sign-in con contraseñas persistentes y una carga segura

Otro método de inicio de sesión con nombre de usuario y contraseña en los grupos de usuarios es el uso del protocolo de contraseña remota segura (SRP). Esta opción envía una prueba del conocimiento de la contraseña (sal y hash de contraseña) que el grupo de usuarios puede verificar. Al no contener información secreta legible en la solicitud a Amazon Cognito, su aplicación es la única entidad que procesa las contraseñas que introducen los usuarios. La autenticación con SRP implica cálculos matemáticos que se pueden realizar mejor con un componente existente que pueda importar a su SDK. Los AWS SDK de bajo nivel exponen las operaciones `InitiateAuth` y las de la `RespondToAuthChallenge` API, pero no calculan los valores de SRP por ti. Higher-level las bibliotecas que implementan SRP, como las bibliotecas Amplify, realizan estos cálculos. Si tu SDK no incluye una implementación de SRP, debes calcular los valores de SRP tú mismo antes de realizar estas operaciones. El SRP se suele implementar en aplicaciones en el cliente, como las aplicaciones móviles. Para obtener más información acerca del protocolo, consulte The Stanford SRP Homepage. Wikipedia también tiene recursos y ejemplos. Hay distintas bibliotecas públicas disponibles para realizar los cálculos de SRP para sus flujos de autenticación.

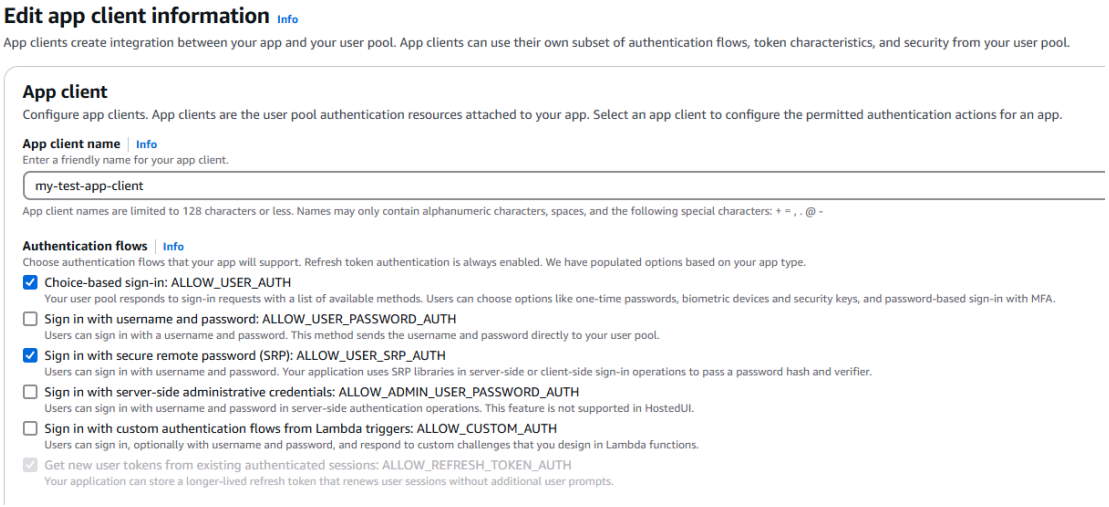
La secuencia inicio/desafío/respuesta de la autenticación de Amazon Cognito valida a los usuarios y sus contraseñas con SRP. Debe configurar su grupo de usuarios y el cliente de aplicación para que admitan la autenticación SRP y, a continuación, implementar la lógica de las solicitudes de inicio de sesión y las respuestas a los desafíos en su aplicación. Sus bibliotecas SRP pueden generar números aleatorios y valores calculados que demuestren a su grupo de usuarios que dispone de la contraseña de un usuario. La aplicación rellena estos valores calculados en los `ChallengeParameters` campos JSON-formatted `AuthParameters` y de los grupos de usuarios de Amazon Cognito, las operaciones de la API y los métodos del SDK para la autenticación.

Activate SRP sign-in

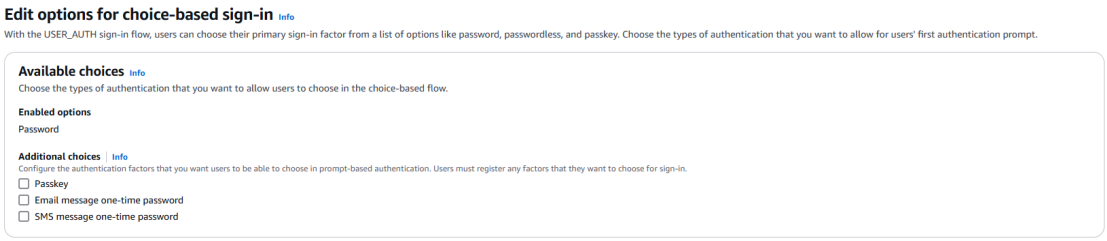
Para activar la autenticación basada en el cliente con nombre de usuario y SRP, configure el cliente de aplicación para que lo permita. En la consola de Amazon Cognito, navegue hasta el menú de clientes de aplicaciones en la configuración de su grupo de usuarios. Para permitir el inicio de sesión con SRP en una aplicación móvil o nativa en el cliente, edite

un cliente de aplicación y seleccione Iniciar sesión con contraseña remota segura (SRP): ALLOW_USER_SRP_AUTH en Flujos de autenticación.

Para activar la autenticación basada en elecciones con nombre de usuario y SRP, edite el cliente de la aplicación y seleccione Choice-based el inicio de sesión: ALLOW_USER_AUTH.



Para comprobar que la autenticación SRP está disponible en tus flujos de autenticación basados en elecciones, navega hasta el menú y consulta la sección de Opciones para el inicio de sesión basado en elecciones. Sign-in Puede iniciar sesión con una autenticación con SRP si la contraseña está visible en Opciones disponibles. La opción de contraseña incluye las variantes de autenticación simple y SRP con nombre de usuario y contraseña.



Configure `ExplicitAuthFlows` con las opciones de autenticación de nombre de usuario y contraseña que prefiera en una solicitud o. CreateUserPoolClientUpdateUserPoolClient

```
"ExplicitAuthFlows": [
   "ALLOW_USER_SRP_AUTH",
   "ALLOW_USER_AUTH"
]
```

En una UpdateUserPoolsolicitud CreateUserPoolo, configúrela `Policies` con los flujos de autenticación basados en opciones que desee admitir. El valor `PASSWORD` en

`AllowedFirstAuthFactors` incluye las opciones de flujo de autenticación con contraseña simple y SRP.

```
"Policies": {
   "SignInPolicy": {
      "AllowedFirstAuthFactors": [
         "PASSWORD",
         "EMAIL_OTP",
         "WEB_AUTHN"
      ]
   }
}
```

Choice-based sign-in with SRP

Para iniciar sesión en una aplicación mediante la autenticación de nombre de usuario y contraseña con SRP, configure el cuerpo de su solicitud de la siguiente manera. AdminInitiateAuthInitiateAuth Esta solicitud de inicio de sesión se realiza correctamente o continúa hasta el siguiente desafío si el usuario actual cumple los requisitos para la autenticación con nombre de usuario y contraseña. De lo contrario, responde con una lista de los desafíos de autenticación de factor principal disponibles. Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PREFERRED_CHALLENGE" : "PASSWORD_SRP",
      "SRP_A" : "[g^a % N]"
   },
   "ClientId": "1example23456789"
}
```

También puede omitir el valor de `PREFERRED_CHALLENGE` y recibir una respuesta que contenga una lista de los factores de inicio de sesión aptos para el usuario.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser"
   },
```

```
   "ClientId": "1example23456789"
}
```

Si no ha enviado un desafío preferido o el usuario enviado no reúne los requisitos para su desafío preferido, Amazon Cognito devolverá una lista de opciones en `AvailableChallenges`. Si `AvailableChallenges` incluye un `ChallengeName` de`PASSWORD_SRP`, puede continuar con la autenticación con una respuesta RespondToAuthChallengeo AdminRespondToAuthChallengeimpugnación en el siguiente formato. Debe pasar un parámetro `Session` que asocie la respuesta al desafío con la respuesta de la API a su solicitud de inicio de sesión inicial. Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

```
{
   "ChallengeName": "PASSWORD_SRP",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "SRP_A" : "[g^a % N]"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response"
}
```

Amazon Cognito responde a las solicitudes de desafío preferente y a las respuestas de desafíos de `PASSWORD_SRP` aptas con un desafío `PASSWORD_VERIFIER`. Su cliente debe completar los cálculos de SRP y responder al desafío en una AdminRespondToAuthChallengesolicitud RespondToAuthChallengeo.

```
{
   "ChallengeName": "PASSWORD_VERIFIER",
   "ChallengeResponses": {
      "PASSWORD_CLAIM_SIGNATURE" : "string",
      "PASSWORD_CLAIM_SECRET_BLOCK" : "string",
      "TIMESTAMP" : "string"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

Si la respuesta a un desafío `PASSWORD_VERIFIER` es correcta, Amazon Cognito emite tokens u otro desafío obligatorio como la autenticación multifactor (MFA).

Client-based sign-in with SRP

La autenticación SRP es más común en la autenticación en el cliente que en el servidor. Sin embargo, puede usar la autenticación SRP con InitiateAuthy. AdminInitiateAuth Para que un usuario inicie sesión en una aplicación, configure el cuerpo de su solicitud `InitiateAuth` o `AdminInitiateAuth` de la siguiente manera. Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

El cliente genera `SRP_A` a partir de un módulo generador N g elevado a la potencia de un entero aleatorio secreto a.

```
{
   "AuthFlow": "USER_SRP_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "SRP_A" : "[g^a % N]"
   },
   "ClientId": "1example23456789"
}
```

Amazon Cognito responde con un desafío `PASSWORD_VERIFIER`. Su cliente debe completar los cálculos de SRP y responder al desafío en una solicitud RespondToAuthChallengeo AdminRespondToAuthChallenge.

```
{
   "ChallengeName": "PASSWORD_VERIFIER",
   "ChallengeResponses": {
      "PASSWORD_CLAIM_SIGNATURE" : "string",
      "PASSWORD_CLAIM_SECRET_BLOCK" : "string",
      "TIMESTAMP" : "string"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

Si la respuesta a un desafío `PASSWORD_VERIFIER` es correcta, Amazon Cognito emite tokens u otro desafío obligatorio como la autenticación multifactor (MFA).

Inicio de sesión sin contraseña con contraseñas de un solo uso

Las contraseñas se pueden perder o robar. Es posible que solo desee comprobar si sus usuarios tienen acceso a una dirección de correo electrónico verificada, un número de teléfono o una aplicación de autenticación. La solución a este problema es el inicio de sesión sin contraseña. Su aplicación puede solicitar a los usuarios que introduzcan su nombre de usuario, dirección de correo electrónico o número de teléfono. A continuación, Amazon Cognito genera una contraseña de un solo uso (OTP), un código que deben confirmar. Un código correcto completa la autenticación.

One-time los flujos de autenticación por contraseña (OTP) no son compatibles con la autenticación multifactor (MFA) requerida en su grupo de usuarios. La autenticación por clave de paso con verificación de usuario puede cumplir con los requisitos de MFA cuando se configura`FactorConfiguration`. `MULTI_FACTOR_WITH_USER_VERIFICATION` Si la MFA es opcional en su grupo de usuarios, los usuarios que hayan activado la MFA no podrán iniciar sesión con un factor OTP como primer factor. Los usuarios que no tengan una preferencia de MFA en un grupo de MFA-optional usuarios pueden iniciar sesión con factores sin contraseña. Para obtener más información, consulte Cosas que debe saber acerca de la MFA de grupos de usuarios.

Cuando un usuario introduce correctamente un código que ha recibido en un mensaje SMS o de correo electrónico como parte de la autenticación sin contraseña, además de autenticar al usuario, el grupo de usuarios marca como verificado el atributo de dirección de correo electrónico o número de teléfono del usuario no verificado. El estado del usuario también cambió de `UNCONFIRMED` a `CONFIRMED`, independientemente de si configuró su grupo de usuarios para verificar automáticamente las direcciones de correo electrónico o los números de teléfono.

Nuevas opciones con inicio de sesión sin contraseña

Al activar la autenticación sin contraseña en el grupo de usuarios, se cambia el funcionamiento de algunos flujos de usuarios.

1. Los usuarios pueden registrarse sin contraseña y elegir un factor sin contraseña al iniciar sesión. También puede crear usuarios sin contraseñas como administrador.
2. Los usuarios que importe con un archivo CSV pueden iniciar sesión inmediatamente sin necesidad de contraseñas. No es necesario que establezcan una contraseña antes de iniciar sesión.
3. Los usuarios que no tengan una contraseña pueden enviar solicitudes de ChangePasswordAPI sin el parámetro. `PreviousPassword`

Inicio de sesión automático con OTP

Los usuarios que se registren y confirmen sus cuentas de usuario mediante OTP con correo electrónico o mensaje SMS pueden iniciar sesión automáticamente con un factor sin contraseña que coincida con su mensaje de confirmación. En la interfaz de usuario con inicio de sesión administrado, los usuarios que confirmen sus cuentas y puedan iniciar sesión con OTP mediante el código de confirmación procederán automáticamente a iniciar sesión por primera vez después de proporcionar el código de confirmación. En tu aplicación personalizada con un AWS SDK, transfiere los siguientes parámetros a una operación InitiateAutho AdminInitiateAuth.

- El `Session` parámetro de la respuesta de la ConfirmSignUpAPI como parámetro de `Session` solicitud.
- Uno AuthFlowde`USER_AUTH`.

Puede pasar un PREFERRED_CHALLENGE de `EMAIL_OTP` o `SMS_OTP`, pero no es obligatorio. El parámetro `Session` proporciona una prueba de autenticación y Amazon Cognito ignora los `AuthParameters` cuando se pasa un código de sesión válido.

La operación de inicio de sesión devuelve la respuesta que indica que la autenticación se ha realizado correctamente AuthenticationResult, sin problemas adicionales si se cumplen las siguientes condiciones.

- El código `Session` es válido y no ha caducado.
- El usuario es apto para el método de autenticación OTP.

Activate passwordless sign-in

Consola

Para activar el inicio de sesión sin contraseña, configure su grupo de usuarios para permitir el inicio de sesión principal con uno o más tipos sin contraseña; luego, configure el cliente de aplicación para permitir el flujo `USER_AUTH`. En la consola de Amazon Cognito, navegue hasta el Sign-inmenú situado en Autenticación en la configuración de su grupo de usuarios. Edite las Opciones para el inicio de sesión basado en opciones y elija Contraseña de un solo uso por mensaje de correo electrónico o Contraseña de un solo uso por mensaje SMS. Puede activar ambas opciones. Guarde los cambios.

Vaya al menú Clientes de aplicación y elija un cliente de aplicación o cree uno nuevo. Seleccione Editar y elija Seleccionar un tipo de autenticación en el inicio de sesión: ALLOW_USER_AUTH.

API/SDK

En la API de grupos de usuarios, `SignInPolicy` configúrela con las opciones sin contraseña adecuadas en una CreateUserPoolsolicitud o. UpdateUserPool

```
"SignInPolicy": {
    "AllowedFirstAuthFactors": [
        "EMAIL_OTP",
        "SMS_OTP"
    ]
}
```

Configure el cliente de la aplicación `ExplicitAuthFlows` con la opción requerida en una solicitud CreateUserPoolClientto UpdateUserPoolClient.

```
"ExplicitAuthFlows": [
   "ALLOW_USER_AUTH"
]
```

Sign in with passwordless

El inicio de sesión sin contraseña no tiene un cliente `AuthFlow` que puedas especificar en y. InitiateAuthAdminInitiateAuth La autenticación OTP solo está disponible en función de la elecciónUSER_AUTH, donde puedes solicitar una opción `AuthFlow` de inicio de sesión preferida o elegir la opción sin contraseña de la del usuario. AvailableChallenges Para que un usuario inicie sesión en una aplicación, configure el cuerpo de su solicitud `InitiateAuth` o `AdminInitiateAuth` de la siguiente manera. Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

En este ejemplo, no sabemos de qué forma quiere iniciar sesión el usuario. Si añadimos un parámetro `PREFERRED_CHALLENGE` y el desafío preferido está disponible para el usuario, Amazon Cognito responde con ese desafío.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser"
   },
   "ClientId": "1example23456789"
}
```

En lugar de eso, puede añadir `"PREFERRED_CHALLENGE": "EMAIL_OTP"` o `"PREFERRED_CHALLENGE": "SMS_OTP"` en `AuthParameters` en este ejemplo. Si el usuario cumple los requisitos para utilizar ese método preferido, el grupo de usuarios envía inmediatamente un código a la dirección de correo electrónico o al número de teléfono del usuario y devuelve `"ChallengeName": "EMAIL_OTP"` o `"ChallengeName": "SMS_OTP"`.

Si no especifica un desafío preferido, Amazon Cognito responde con un parámetro `AvailableChallenges`.

```
{
   "AvailableChallenges": [
      "EMAIL_OTP",
      "SMS_OTP",
      "PASSWORD"
    ],
   "Session": "[Session ID]"
}
```

Este usuario puede iniciar sesión sin contraseña con la OTP con mensaje de correo electrónico, la OTP con mensaje SMS y la combinación de nombre de usuario y contraseña. La aplicación puede pedirle al usuario su elección o realizar una elección en función de la lógica interna. A continuación, procede con una solicitud RespondToAuthChallengeo AdminRespondToAuthChallengesolicitud en la que se selecciona el desafío. Supongamos que el usuario desea completar la autenticación sin contraseña con una OTP con mensaje de correo electrónico.

```
{
   "ChallengeName": "SELECT_CHALLENGE",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "ANSWER" : "EMAIL_OTP"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

Amazon Cognito responde con un desafío `EMAIL_OTP` y envía un código a la dirección de correo electrónico verificada del usuario. A continuación, su aplicación debe volver a responder a este desafío.

Esta también sería la siguiente respuesta al desafío si ha solicitado `EMAIL_OTP` como `PREFERRED_CHALLENGE`.

```
{
   "ChallengeName": "EMAIL_OTP",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "EMAIL_OTP_CODE" : "123456"
   },
   "ClientId": "1example23456789",
   "Session": "[Session ID from the previous response]"
}
```

Inicio de sesión sin contraseña con claves de acceso WebAuthn

Las claves de acceso son seguras e imponen un nivel de esfuerzo relativamente bajo a los usuarios. El inicio de sesión con clave de acceso usa autenticadores, dispositivos externos que los usuarios pueden usar para la autenticación. Las contraseñas habituales exponen a los usuarios a vulnerabilidades, como la suplantación de identidad, la adivinación de contraseñas y el robo de credenciales. Con las claves de acceso, su aplicación puede beneficiarse de medidas de seguridad avanzadas en los teléfonos móviles y otros dispositivos conectados a los sistemas de información o integrados en ellos. Un flujo de trabajo de inicio de sesión con clave de acceso común comienza con una llamada al dispositivo que invoca al administrador de contraseñas o credenciales, por ejemplo, el llavero de iOS o el administrador de contraseñas de Google Chrome. El administrador de credenciales del dispositivo les pide que seleccionen una clave de acceso y la autoricen con una credencial existente o un mecanismo de desbloqueo del dispositivo. Los teléfonos modernos tienen escáneres faciales, escáneres de huellas digitales, patrones de desbloqueo y otros mecanismos, algunos de los cuales cumplen simultáneamente los principios de autenticación sólida de algo que ya sabes y algo que tienes. En el caso de la autenticación mediante claves de acceso biométricas, las claves de acceso son un método del tipo algo que eres.

Quizá quiera sustituir las contraseñas por la autenticación mediante huella digital, la identificación facial o la clave de seguridad. Se trata de una clave de paso o autenticación. WebAuthn Es habitual que los desarrolladores de aplicaciones permitan a los usuarios inscribir un dispositivo biométrico después de iniciar sesión por primera vez con una contraseña. Con los grupos de usuarios de Amazon Cognito, su aplicación puede configurar esta opción de inicio de sesión para los usuarios. La autenticación con clave de paso puede cumplir con los requisitos de autenticación multifactor (MFA) cuando su grupo de usuarios se haya establecido en. `FactorConfiguration`

`MULTI_FACTOR_WITH_USER_VERIFICATION` En esta configuración, la autenticación con clave de paso con verificación de usuario cuenta como autenticación multifactorial.

One-time los flujos de autenticación por contraseña (OTP) no son compatibles con la autenticación multifactor (MFA) requerida en su grupo de usuarios. La autenticación por clave de paso con verificación de usuario puede cumplir con los requisitos de MFA cuando se configura`FactorConfiguration`. `MULTI_FACTOR_WITH_USER_VERIFICATION` Si la MFA es opcional en su grupo de usuarios, los usuarios que hayan activado la MFA no podrán iniciar sesión con un factor OTP como primer factor. Los usuarios que no tengan una preferencia de MFA en un grupo de MFA-optional usuarios pueden iniciar sesión con factores sin contraseña. Para obtener más información, consulte Cosas que debe saber acerca de la MFA de grupos de usuarios.

¿Qué son las claves de acceso?

Las claves de acceso simplifican la experiencia del usuario, ya que eliminan la necesidad de recordar contraseñas complejas o introducir contraseñas de un solo uso (OTP). Las claves de acceso se basan en WebAuthn los estándares CTAP2 redactados por el World Wide Web Consortium (W3C) y la Alianza FIDO (Fast Identity Online). Los navegadores y las plataformas implementan estos estándares, proporcionan API para que las aplicaciones web o móviles inicien un proceso de registro o autenticación de claves de acceso y aportan una interfaz de usuario para que el usuario seleccione un autenticador de clave de acceso e interactúe con él.

Cuando un usuario registra un autenticador en un sitio web o una aplicación, el autenticador crea un key pair público-privado. WebAuthn los navegadores y las plataformas envían la clave pública al back-end de la aplicación, del sitio web o la aplicación. El autenticador guarda la clave privada, los ID de las claves y los metadatos sobre el usuario y la aplicación. Cuando el usuario quiere autenticarse en la aplicación registrada con su autenticador registrado, la aplicación genera un desafío aleatorio. La respuesta a este desafío es la firma digital del desafío generada con la clave privada del autenticador para esa aplicación y usuario, así como los metadatos relevantes. El navegador o la plataforma de la aplicación reciben la firma digital y la pasan al backend de la aplicación. A continuación, la aplicación valida la firma con la clave pública almacenada.

> **Note**
>
> La aplicación no recibe ningún secreto de autenticación que los usuarios proporcionen a su autenticador, ni recibe información sobre la clave privada.

A continuación, se muestran algunos ejemplos y capacidades de los autenticadores actualmente en el mercado. Un autenticador puede cumplir con alguna de estas categorías o con todas ellas.

- Algunos autenticadores verifican al usuario con factores como un PIN, una entrada biométrica con un rostro o huella digital o un código de acceso antes de conceder dicho acceso, lo que garantiza que solo el usuario legítimo pueda autorizar las acciones. Otros autenticadores no tienen ninguna función de verificación de usuario y algunos pueden omitir la verificación de usuario cuando una aplicación no la requiere.
- Algunos autenticadores, como los tokens YubiKey de hardware, son portátiles. Se comunican con los dispositivos a través de conexiones USB, Bluetooth o NFC. Algunos autenticadores son locales y están vinculados a una plataforma, como Windows Hello en un ordenador o Face ID en un iPhone. El usuario puede transportar un autenticador vinculado a un dispositivo si es lo suficientemente pequeño, como un dispositivo móvil. A veces, los usuarios pueden conectar su autenticador de hardware a muchas plataformas diferentes mediante comunicación inalámbrica. Por ejemplo, los usuarios de los navegadores de escritorio pueden usar su teléfono inteligente como autenticador de clave de acceso cuando escanean un código QR.
- Algunas claves de acceso vinculadas a la plataforma se sincronizan con la nube para poder utilizarlas desde varios lugares. Por ejemplo, las claves de acceso de Face ID de los iPhones sincronizan los metadatos de las claves con las cuentas de Apple de los usuarios en su llavero de iCloud. Estas claves de acceso permiten una autenticación perfecta en todos los dispositivos Apple, en lugar de requerir que los usuarios registren cada dispositivo de forma independiente. Software-based Las aplicaciones autenticadoras como 1Password, Dashlane y Bitwarden sincronizan las claves de acceso en todas las plataformas en las que el usuario ha instalado la aplicación.

En WebAuthn terminología, los sitios web y las aplicaciones son partes de confianza. Cada clave de acceso está asociada a un identificador de usuario específico, un identificador unificado que representa los sitios web o las aplicaciones que aceptan la autenticación con clave de acceso. Los desarrolladores deben seleccionar cuidadosamente su ID de actor de confianza para tener el alcance de autenticación adecuado. Un identificador típico de la parte que confía es el nombre de dominio raíz de un servidor web. Una clave de acceso con esta especificación de ID de actor de confianza puede autenticar ese dominio y sus subdominios. Los navegadores y las plataformas deniegan la autenticación con clave de paso cuando la URL del sitio web al que el usuario quiere acceder no coincide con el ID de la persona que confía en él. Del mismo modo, en el caso de las aplicaciones móviles, solo se puede usar una clave de acceso si la ruta de la aplicación está presente en los

archivos de asociación `.well-known` que la aplicación pone a disposición en la ruta indicada por el ID del actor de confianza.

Las claves de acceso son reconocibles. Un navegador o una plataforma pueden reconocerlas y utilizarlas automáticamente sin necesidad de que el usuario introduzca un nombre de usuario. Cuando un usuario visita un sitio web o una aplicación que admite la autenticación con clave de acceso, puede seleccionarlas de una lista de claves que el navegador o la plataforma ya conocen, o puede escanear un código QR.

¿Cómo implementa Amazon Cognito la autenticación con clave de acceso?

Las claves de acceso son una característica opcional que está disponible en todos los planes de características, con la excepción de Lite. Solo está disponible en el flujo de autenticación basado en opciones. Con el inicio de sesión administrado, Amazon Cognito gestiona la lógica de la autenticación con clave de acceso. También puede usar la API de grupos de usuarios de Amazon Cognito en los AWS SDK para realizar la autenticación con clave de acceso en el backend de la aplicación.

Amazon Cognito reconoce las claves de acceso creadas con uno de los dos algoritmos criptográficos asimétricos, ES256(-7) y RS256(-257). La mayoría de los autenticadores admiten ambos algoritmos. De forma predeterminada, los usuarios pueden configurar cualquier tipo de autenticador, por ejemplo, tokens de hardware, teléfonos móviles inteligentes y aplicaciones de autenticación de software. Amazon Cognito no admite actualmente la aplicación de atestaciones.

En su grupo de usuarios, puede configurar la verificación de usuario como preferida u obligatoria. Esta configuración se establece de forma predeterminada en las solicitudes de API que no proporcionan un valor, y se selecciona de forma predeterminada en la consola de Amazon Cognito. Al configurar la verificación de usuarios como preferida, los usuarios pueden configurar autenticadores que no tienen la capacidad de verificación de usuarios, y las operaciones de registro y autenticación se pueden realizar correctamente sin la verificación del usuario. Para exigir la verificación de los usuarios en el registro y la autenticación con clave de acceso, cambie esta configuración a obligatoria.

El identificador del actor de confianza (RP) que establezca en la configuración de la clave de acceso es una decisión importante. Si no especifica lo contrario y la versión de marca del dominio es el inicio de sesión administrado, su grupo de usuarios esperará de forma predeterminada que el nombre de su dominio personalizado sea el ID del RP. Si no tiene un dominio personalizado y no especifica lo contrario, su grupo de usuarios utilizará de forma predeterminada un ID de RP de su dominio de prefijo. También puede configurar su ID de RP para que sea cualquier nombre de dominio que

no figure en la lista de sufijos públicos (PSL). La entrada de su ID de RP se aplica al registro y la autenticación con clave de acceso en el inicio de sesión administrado y en la autenticación del SDK. Passkey solo funciona en aplicaciones móviles cuando Amazon Cognito puede localizar `.well-known` un archivo de asociación con su ID de RP como dominio. Como práctica recomendada, determine y establezca el valor del ID del actor de confianza antes de que su sitio web o aplicación estén disponibles públicamente. Si cambia su ID de RP, los usuarios deberán volver a registrarse con el nuevo ID de RP.

Cada usuario puede registrar hasta 20 claves de acceso. Solo pueden registrar una clave de acceso después de haber iniciado sesión en su grupo de usuarios al menos una vez. El inicio de sesión administrado supone un esfuerzo considerable para el registro de la clave de acceso. Al habilitar la autenticación con clave de acceso para un grupo de usuarios y un cliente de aplicación, el grupo de usuarios con un dominio de inicio de sesión administrado recuerda a los usuarios finales que deben registrar una clave de acceso después de crear una nueva cuenta de usuario. También puede invocar los navegadores de los usuarios en cualquier momento para dirigirlos a una página de inicio de sesión administrado para el registro de la clave de acceso. Los usuarios deben proporcionar un nombre de usuario antes de que Amazon Cognito pueda iniciar la autenticación con clave de acceso. El inicio de sesión administrado lo gestiona automáticamente. La página de inicio de sesión solicita un nombre de usuario, valida que el usuario tenga registrada al menos una clave de acceso y, a continuación, solicita el inicio de sesión con la clave de acceso. Del mismo modo, SDK-based las aplicaciones deben solicitar un nombre de usuario y proporcionarlo en la solicitud de autenticación.

Cuando configuras la autenticación de grupos de usuarios con claves de paso y tienes un dominio personalizado y un dominio de prefijo, el ID de RP toma como valor predeterminado el nombre de dominio completo (FQDN) de tu dominio personalizado. Para configurar un dominio de prefijo como ID de RP en la consola de Amazon Cognito, elimine su dominio personalizado o introduzca el FQDN del dominio de prefijo como dominio. Third-party

Activate passkey sign-in

Consola

Para activar el inicio de sesión con clave de acceso, configure su grupo de usuarios para permitir el inicio de sesión principal con uno o más tipos sin contraseña; luego, configure el cliente de aplicación para permitir el flujo `USER_AUTH`. En la consola de Amazon Cognito, navegue hasta el Sign-inmenú situado en Autenticación en la configuración de su grupo de usuarios. Edite las Opciones para el inicio de sesión basado en opciones y añada Clave de acceso a la lista de opciones disponibles.

Vaya al menú Métodos de autenticación y edite la clave de acceso.

- La verificación de usuario es la configuración que determina si su grupo de usuarios requiere dispositivos con clave de acceso que comprueben además si el usuario actual está autorizado a utilizar una clave de acceso. Para animar a los usuarios a configurar un dispositivo con la verificación de usuario, pero sin exigirla, seleccione Preferente. Para admitir únicamente dispositivos con verificación de usuario, seleccione Obligatorio. Para obtener más información, consulte la sección User verification en w3.org.
- El dominio para el identificador de la parte de confianza es el identificador que su aplicación transfiere en las solicitudes de registro de claves de paso de los usuarios. Define el destino de la relación de confianza con el emisor de las claves de acceso de los usuarios. El identificador de la persona que confía puede ser uno de los siguientes:

  dominio de Cognito

  El dominio de prefijo de Amazon Cognito de su grupo de usuarios.

  Dominio personalizado

  El dominio personalizado de su grupo de usuarios.

  Dominio Third-party

  El dominio de las aplicaciones que no utilizan las páginas de inicio de sesión administrado para grupos de usuarios. Esta configuración suele estar asociada a grupos de usuarios que no tienen un dominio y realizan la autenticación con un AWS SDK y la API de grupos de usuarios en el back-end.

Vaya al menú Clientes de aplicación y elija un cliente de aplicación o cree uno nuevo. Seleccione Editar y, en Flujos de autenticación, elija Seleccionar un tipo de autenticación en el inicio de sesión: ALLOW_USER_AUTH.

API/SDK

En la API de grupos de usuarios, `SignInPolicy` configúrela con las opciones de clave de paso adecuadas en una UpdateUserPoolsolicitud CreateUserPoolo. La opción `WEB_AUTHN` para la autenticación con clave de acceso debe ir acompañada de al menos otra opción. El registro de la clave de acceso requiere una sesión de autenticación existente.

```
"SignInPolicy": {
    "AllowedFirstAuthFactors": [
```

```
        "PASSWORD",
        "WEB_AUTHN"
    ]
}
```

Configure su preferencia de verificación de usuario y su ID de RP en el `WebAuthnConfiguration` parámetro de una solicitud. SetUserPoolMfaConfig El `RelyingPartyId`, el objetivo previsto de los resultados de la autenticación con clave de acceso, puede ser el prefijo de su grupo de usuarios, un dominio personalizado o un dominio que usted elija.

```
"WebAuthnConfiguration": {
   "RelyingPartyId": "example.auth.us-east-1.amazoncognito.com",
   "UserVerification": "preferred",
   "FactorConfiguration": "SINGLE_FACTOR"
}
```

Configura el cliente de tu aplicación `ExplicitAuthFlows` con la opción requerida en una solicitud CreateUserPoolClientto UpdateUserPoolClient.

```
"ExplicitAuthFlows": [
   "ALLOW_USER_AUTH"
]
```

Register a passkey (managed login)

El inicio de sesión administrado gestiona el registro de las claves de acceso por parte de los usuarios. Cuando la autenticación con clave de paso está activa en su grupo de usuarios, Amazon Cognito les pide a los usuarios que configuren una clave de paso al registrar una nueva cuenta de usuario.

Amazon Cognito no solicita a los usuarios que configuren una clave de acceso si ya se han registrado y no han configurado una o si usted creó su cuenta como administrador. Los usuarios de este estado deben iniciar sesión con otro factor, como una contraseña o una OTP sin contraseña, antes de poder registrar una clave de acceso.

Cómo registrar una clave de acceso

1. Dirija al usuario a su página de inicio de sesión.

```
https://auth.example.com/oauth2/authorize/?
client_id=1example23456789&response_type=code&scope=email+openid
+phone&redirect_uri=https%3A%2F%2Fwww.example.com
```

2. Procese el resultado de la autenticación del usuario. En este ejemplo, Amazon Cognito los redirige a `www.example.com` con un código de autorización que la aplicación intercambia por tokens.
3. Dirija al usuario a su página de registro y clave de acceso. El usuario dispondrá de una cookie del navegador que mantendrá su sesión de inicio de sesión. La URL de la clave de acceso toma los parámetros `client_id` y `redirect_uri`. Amazon Cognito solo permite el acceso a esta página a los usuarios autenticados. Inicie sesión en su usuario con una contraseña, una OTP con correo electrónico o una OTP con SMS; luego, invoque una URL que coincida con el siguiente patrón.

   También puede añadir otros parámetros Autorizar punto de conexión a esta solicitud, como `response_type` y `scope`.

```
https://auth.example.com/passkeys/add?
client_id=1example23456789&redirect_uri=https%3A%2F%2Fwww.example.com
```

Register a passkey (SDK)

Las credenciales de la clave de paso se registran con los metadatos de un objeto. PublicKeyCreationOptions Puede generar este objeto con las credenciales de un usuario que haya iniciado sesión y presentarlas en una solicitud de API al emisor de la clave de acceso. El emisor devuelve un objeto RegistrationResponseJSON que confirma el registro de la clave de paso.

Para iniciar el proceso de registro de la clave de acceso, inicie sesión con un usuario que tenga una opción de inicio de sesión existente. Autoriza la solicitud de StartWebAuthnRegistrationAPI autorizada por el token con el token de acceso del usuario actual. A continuación, se muestra el cuerpo de una solicitud `GetWebAuthnRegistrationOptions` de ejemplo.

```
{
   "AccessToken": "eyJra456defEXAMPLE"
}
```

La respuesta de su grupo de usuarios contiene el objeto `PublicKeyCreationOptions`. Presente este objeto en una solicitud de API al emisor del usuario. Proporciona información como la clave pública y el identificador del actor de confianza. El emisor responde con un objeto. `RegistrationResponseJSON`

Presente la respuesta de registro en una solicitud de CompleteWebAuthnRegistrationAPI, nuevamente autorizada con el token de acceso del usuario. Cuando el grupo de usuarios responde con una respuesta HTTP 200 con el cuerpo vacío, se registra la clave de acceso del usuario.

Sign in with a passkey

El inicio de sesión sin contraseña no tiene ninguna `AuthFlow` que puedas especificar en y. InitiateAuthAdminInitiateAuth En su lugar, debe declarar un `AuthFlow` de `USER_AUTH` y solicitar una opción de inicio de sesión o elegir la opción sin contraseña en la respuesta de su grupo de usuarios. Para que un usuario inicie sesión en una aplicación, configure el cuerpo de su solicitud `InitiateAuth` o `AdminInitiateAuth` de la siguiente manera. Este conjunto de parámetros es el mínimo necesario para iniciar sesión. Hay parámetros adicionales disponibles.

En este ejemplo, sabemos que el usuario quiere iniciar sesión con una clave de acceso y añadimos un parámetro `PREFERRED_CHALLENGE`.

```
{
   "AuthFlow": "USER_AUTH",
   "AuthParameters": {
      "USERNAME" : "testuser",
      "PREFERRED_CHALLENGE" : "WEB_AUTHN"
   },
   "ClientId": "1example23456789"
}
```

Amazon Cognito responde con un desafío `WEB_AUTHN`. Su aplicación debe responder a este desafío. Inicie una solicitud de inicio de sesión con el proveedor de la clave de acceso del usuario. Devuelve un objeto JSONAuthenticationResponse.

```
{
   "ChallengeName": "WEB_AUTHN",
   "ChallengeResponses": {
      "USERNAME" : "testuser",
      "CREDENTIAL" : "{AuthenticationResponseJSON}"
```

```
    },
    "ClientId": "1example23456789",
    "Session": "[Session ID from the previous response]"
}
```

MFA después del inicio de sesión

Puede configurar los usuarios que completen el inicio de sesión con un flujo de nombre de usuario y contraseña para que se les solicite una verificación adicional con una contraseña de un solo uso desde un mensaje de correo electrónico, un mensaje SMS o una aplicación generadora de códigos. La MFA es diferente del inicio de sesión sin contraseña con contraseñas de un solo uso. Sin embargo, las claves de paso con verificación de usuario pueden cumplir los requisitos de MFA como primer factor cuando se `FactorConfiguration` configura `MULTI_FACTOR_WITH_USER_VERIFICATION` como en su grupo de usuarios. `WebAuthnConfiguration` Las claves de paso no se pueden utilizar como un segundo factor para iniciar sesión con contraseña. Para los flujos basados en contraseñas, la MFA en los grupos de usuarios es un modelo de desafío-respuesta en el que el usuario primero demuestra que conoce la contraseña y, a continuación, demuestra que tiene acceso al dispositivo de segundo factor registrado.

Recursos de implementación

- Adición de MFA a un grupo de usuarios.

Tokens de actualización

Si quiere mantener a los usuarios conectados sin tener que volver a introducir sus credenciales, los tokens de actualización son la herramienta de la que dispone su aplicación para conservar la sesión de un usuario. Las aplicaciones pueden presentar tokens de actualización a su grupo de usuarios e intercambiarlos por nuevos tokens de ID y acceso. Con la actualización de los tokens, puede asegurarse de que un usuario que haya iniciado sesión siga activo, obtener información de atributos actualizada y actualizar los derechos de control de acceso sin la intervención del usuario.

Recursos de implementación

- Tokens de actualización

Autenticación personalizada

Es posible que desee configurar un método de autenticación para sus usuarios que no aparezca en esta lista. Puede hacerlo con una autenticación personalizada con activadores Lambda. En una secuencia de funciones de Lambda, Amazon Cognito genera un desafío, formula una pregunta que los usuarios deban responder, comprueba la precisión de la respuesta y, a continuación, determina si debe emitirse otro desafío. Las preguntas y respuestas pueden incluir preguntas de seguridad, solicitudes a un servicio de CAPTCHA, solicitudes a una API de servicio de MFA externa o todas ellas en secuencia.

Recursos de implementación

- Desencadenadores de Lambda de desafío de autenticación personalizado

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 challenge/response basado en activadores. AWS Lambda

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 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 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 con un desencadenador de Lambda para la migración de usuarios.

Modelos de autorización para la autenticación de API y SDK

Cuando comience a desarrollar su aplicación con la autenticación de grupos de usuarios, debe decidir el modelo de autorización de la API que mejor se adapte a su tipo de aplicación. Un modelo de autorización es un sistema que proporciona autorización para realizar solicitudes con los componentes de autenticación de las integraciones de la API y el SDK de los grupos de usuarios de Amazon Cognito. Amazon Cognito tiene tres modelos de autorización: pública y IAM-authorized autorizada por token.

En el caso de IAM-authorized las solicitudes, la autorización proviene de la firma de un conjunto de credenciales de AWS IAM en el encabezado de la `Authorization` solicitud. En el caso de las aplicaciones en el servidor, este método protege las operaciones de autenticación con la autorización de IAM. En el caso de las solicitudes de autenticación públicas (no autenticadas), no se requiere autorización. Esto es adecuado para las aplicaciones en el cliente distribuidas a los usuarios. En el caso de las operaciones autorizadas por un token, que normalmente se implementan en combinación con operaciones públicas, la autorización proviene de un token de sesión o un token de acceso incluido en el encabezado de `Authorization` de la solicitud. La autenticación de Amazon Cognito normalmente requiere que implemente dos o más operaciones de API en orden; las operaciones de API que utilice dependen de las características de la aplicación. Los clientes públicos, en los que la aplicación se distribuye a los usuarios, utilizan operaciones públicas, en las que las solicitudes de inicio de sesión no requieren autorización. Token-authorized las operaciones continúan la sesión de los usuarios en las aplicaciones públicas. Server-side los clientes, donde la lógica de la aplicación está alojada en un sistema remoto, protegen las operaciones de autenticación con la autorización de IAM para las solicitudes de inicio de sesión. Los siguientes pares de operaciones de la API y sus correspondientes métodos del SDK se corresponden con los modelos de autorización disponibles.

Cada operación de autenticación pública tiene algún tipo de equivalente en el servidor, por ejemplo, y. UpdateUserAttributesAdminUpdateUserAttributes Si bien las operaciones en el cliente son iniciadas por el usuario y requieren confirmación, las operaciones en el servidor dan por sentado que el cambio lo ha realizado un administrador del grupo de usuarios, así que los cambios se aplican de forma inmediata. En este ejemplo, Amazon Cognito envía un mensaje con un código de confirmación al usuario y el token de acceso del usuario autoriza una VerifyUserAttributesolicitud que envía el código. La aplicación en el servidor puede establecer inmediatamente el valor de cualquier atributo,

aunque hay que tener en cuenta algunas consideraciones especiales a la hora de cambiar el valor de las direcciones de correo electrónico y los números de teléfono cuando se utilizan para iniciar sesión.

Para comparar la autenticación de la API y ver una lista completa de las operaciones de la API y sus modelos de autorización, consulte Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

Client-side (public) authentication

A continuación se muestra una secuencia típica de solicitudes en una aplicación en el cliente.

1. La InitiateAuthoperación pública envía las credenciales principales, como un nombre de usuario y una contraseña.
2. La RespondToAuthChallengeoperación autorizada por el token envía un token de sesión a partir de la `InitiateAuth` respuesta y la respuesta a un desafío, por ejemplo, MFA. La autorización del token de sesión indica las solicitudes que forman parte de ciclos de autenticación que aún no se han completado.
3. La ConfirmDeviceoperación autorizada con el token envía un token de acceso y realiza la operación de escritura consistente en añadir un dispositivo recordado al perfil del usuario. La autorización del token de acceso indica las solicitudes que se destinan a operaciones de autoservicio del usuario una vez que este haya completado la autenticación.

Para obtener más información, consulte Client-side opciones de autenticación y Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

Server-side authentication

A continuación, puede ver una secuencia típica de solicitudes de una operación en el servidor. Cada solicitud tiene un encabezado de autorización AWS Signature Version 4 firmado con las credenciales de la máquina de IAM emitidas hacia el servidor de aplicaciones.

1. La AdminInitiateAuthoperación envía las credenciales principales, como un nombre de usuario y una contraseña.
2. AdminRespondToAuthChallengela operación envía la respuesta a un desafío, por ejemplo, MFA.
3. La AdminUpdateDeviceStatusoperación establece la clave del dispositivo a partir de la `AdminInitiateAuth` respuesta tal como se recuerda.

Para obtener más información, consulte Server-side opciones de autenticación y Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

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.

Temas

- Server-side opciones de autenticación
- Client-side opciones de autenticación
- Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado
- Lista de operaciones de API agrupadas por modelo de autorización

Server-side opciones de autenticación

Las aplicaciones web y otras aplicaciones del lado del servidor implementan la autenticación en un servidor remoto que un cliente carga en una aplicación de pantalla remota, como un navegador o una sesión SSH. Server-side las aplicaciones suelen tener las siguientes características.

- Están integrados en una aplicación instalada en un servidor en lenguajes como Java, Ruby o Node.js.
- Se conectan a clientes de aplicación de grupos de usuarios que pueden tener un secreto de cliente, denominados clientes confidenciales.
- Tienen acceso a AWS las credenciales.
- Para la autenticación, invocan el inicio de sesión gestionado o utilizan IAM-authorized operaciones en la API de grupos de usuarios con un AWS SDK.
- Atienden a clientes internos y pueden atender a clientes públicos.

Server-side Las operaciones con la API de grupos de usuarios pueden utilizar contraseñas, contraseñas de un solo uso o claves de acceso como factor de inicio de sesión principal. 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 realiza una solicitud a la API. AdminInitiateAuth Esta operación requiere AWS credenciales con permisos que incluyen `cognito-idp:AdminInitiateAuth` y. `cognito-idp:AdminRespondToAuthChallenge` La operación devuelve el resultado de la autenticación o el desafío requeridos.
- Cuando la aplicación recibe una impugnación, realiza una solicitud a la AdminRespondToAuthChallengeAPI. La operación de la API `AdminRespondToAuthChallenge` también requiere credenciales de AWS .

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.

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**
>
> Como las implementaciones de administración interna utilizan el flujo de autenticación de administrador, este flujo no admite dispositivos recordados. Si el seguimiento de dispositivos está activado, la autenticación de administrador se realiza correctamente, pero las llamadas de actualización del token de acceso fallan.

Client-side opciones de autenticación

Las aplicaciones móviles y otros tipos de aplicaciones en el cliente se instalan en los dispositivos de los usuarios y ejecutan la lógica de autenticación y la interfaz de usuario de forma local. Por lo general, tienen las siguientes características.

- Están integradas en lenguajes como React native, Flutter y Swift, y se implementan en los dispositivos de los usuarios.
- Se conectan a clientes de aplicación de grupos de usuarios que no tienen un secreto de cliente, denominados clientes públicos.
- No tienen acceso a AWS las credenciales que autorizarían las solicitudes de IAM-authorized API.

- Utilizan el inicio de sesión gestionado para la autenticación o utilizan operaciones públicas y autorizadas por token en la API de grupos de usuarios con un SDK. AWS
- Atienden a clientes públicos y permiten a cualquiera registrarse e iniciar sesión.

Client-side Las operaciones con la API de grupos de usuarios pueden utilizar contraseñas, contraseñas de un solo uso o claves de acceso como factor de inicio de sesión principal. 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).

Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado

Los grupos de usuarios de Amazon Cognito son una combinación de varias tecnologías de autenticación. Dependen de proveedores de identidad externos ()IdPs. Son IdPs para aplicaciones que implementan la autenticación con los SDK de OpenID Connect (OIDC). Proporcionan autenticación como emisores de tokens web JSON (JWT) de forma similar a la autenticación OIDC,

pero mediante métodos de API que forman parte de los AWS SDK. También pueden ser puntos de entrada seguros a sus aplicaciones.

Para el registro, el inicio de sesión y la administración de usuarios de su grupo de usuarios, tiene dos opciones.

1. Sus páginas de inicio de sesión administrado y la interfaz de usuario alojada clásica incluyen los puntos de conexión interactivos para el usuario de inicio de sesión administrado y los puntos de conexión de federación, que gestionan los IdP y los roles de los actores de confianza. 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 del inicio de sesión administrado.

   Los otros puntos finales del grupo de usuarios facilitan la autenticación con proveedores de identidad de terceros (). IdPs Los servicios que se prestan incluyen los siguientes.

   a. Service-provider llame a los puntos de enlace para obtener las reclamaciones autenticadas de sus IdPs, como y. `saml2/idpresponse` `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 `/.well-known/jwks.json`. La aplicación usa estos puntos de conexión cuando verifica los tokens o recupera los datos del perfil de usuario con las bibliotecas de desarrollador de OIDC u OAuth 2.0.

2. La API de grupos de usuarios de Amazon Cognito es un conjunto de herramientas para la aplicación web o móvil que se emplea para autenticar a usuarios tras recopilar información de inicio de sesión en su propio frontend personalizado. 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 añade costes a tu factura. AWS

c. Un token de actualización que autoriza las solicitudes de nuevos identificadores y tokens de acceso, además de actualizar la identidad del usuario y las propiedades de control de acceso.

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 vinculas su identidad federada a un usuario local en una solicitud de AdminLinkProviderForUserAPI, este podrá iniciar sesión con la API de 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.

1. Crea y configura sus recursos de grupos de usuarios de Amazon Cognito. Por ejemplo, puedes crear grupos de usuarios, añadir AWS Lambda activadores y configurar el dominio del grupo de usuarios que aloja tus páginas de inicio de sesión gestionadas.
2. 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. La aplicación envía una solicitud de SignUpAPI y crea un nuevo usuario en el 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. Tu aplicación envía una solicitud de ConfirmSignUpAPI 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. Tu aplicación envía una solicitud de InitiateAuthAPI 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 el inicio de sesión administrado, consulte Referencia de los puntos de conexión de grupos de usuarios y del inicio de sesión administrado.

Sus usuarios federados pueden iniciar sesión en el inicio de sesión administrado y seleccionar su IdP, o bien puede omitir el inicio de sesión administrado y enviar a sus usuarios directamente a su IdP para que inicien 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.

Ejemplo de escenario con páginas de inicio de sesión administrado

1. El usuario selecciona el botón "Create an account" (Crear una cuenta) que ha creado en su aplicación.
2. El inicio de sesión administrado 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 autenticación de Apple. Su usuario inicia sesión y decide autorizar a Amazon Cognito a leer la información de su 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. La biblioteca OIDC de su aplicación intercambia el código de autorización con el Punto de conexión de token y almacena un token de identificación, un token de acceso y un token de actualización emitidos por el grupo de usuarios. Su aplicación usa las bibliotecas OIDC para administrar los tokens de usuarios y mantener una sesión persistente para ese usuario.

La API de grupos de usuarios y las páginas de inicio de sesión administrado admiten una variedad de 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.

Lista de operaciones de API agrupadas por modelo de autorización

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 IAM-authorized operaciones de administración y IAM-authorized operaciones de usuario.

IAM-authorized operaciones de administración

IAM-authorized las operaciones de administración modifican y visualizan la configuración del grupo de usuarios y el cliente de la aplicación, como lo haría en el Consola de administración de AWS.

Por ejemplo, para modificar su grupo de usuarios en una solicitud de UpdateUserPoolAPI, debe presentar AWS las credenciales y los permisos de IAM para actualizar el recurso.

Para autorizar estas solicitudes en el AWS Command Line Interface (AWS CLI) o en un AWS SDK, configura tu entorno con variables de entorno o con una configuración de cliente que añada credenciales de IAM a tu solicitud. Para obtener más información, consulte Acceder AWS con sus AWS credenciales en. 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 AWS credenciales que inserte en el encabezado de la solicitud. Para obtener más información, consulta Firmar solicitudes de AWS API.

IAM-authorized operaciones de gestión

AddCustomAttributes

IAM-authorized operaciones de gestión

CreateGroup

CreateIdentityProvider

CreateResourceServer

CreateUserImportJob

CreateUserPool

CreateUserPoolClient

CreateUserPoolDomain

DeleteGroup

DeleteIdentityProvider

DeleteResourceServer

DeleteUserPool

DeleteUserPoolClient

DeleteUserPoolDomain

DescribeIdentityProvider

DescribeResourceServer

DescribeRiskConfiguration

DescribeUserImportJob

DescribeUserPool

DescribeUserPoolClient

DescribeUserPoolDomain

GetCSVHeader

IAM-authorized operaciones de gestión

GetGroup

GetIdentityProviderByIdentifier

GetSigningCertificate

GetUICustomization

GetUserPoolMfaConfig

ListGroups

ListIdentityProviders

ListResourceServers

ListTagsForResource

ListUserImportJobs

ListUserPoolClients

ListUserPools

ListUsers

ListUsersInGroup

SetRiskConfiguration

SetUICustomization

SetUserPoolMfaConfig

StartUserImportJob

StopUserImportJob

TagResource

UntagResource

IAM-authorized operaciones de gestión

UpdateGroup

UpdateIdentityProvider

UpdateResourceServer

UpdateUserPool

UpdateUserPoolClient

UpdateUserPoolDomain

IAM-authorized operaciones de usuario

IAM-authorized operaciones de usuario: registrarse, iniciar sesión, administrar las credenciales de sus usuarios, modificarlos y verlos.

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, el servidor puede incluir AWS las credenciales en una solicitud de AdminCreateUserAPI. 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 el SDK AWS CLI o en un AWS SDK, configura el entorno de la aplicación del lado del servidor con variables de entorno o con una configuración de cliente que añada credenciales de IAM a tu solicitud. Para obtener más información, consulte Acceder AWS con sus AWS credenciales en. 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 AWS credenciales que inserte en el encabezado de la solicitud. Para obtener más información, consulta Firmar solicitudes de AWS API.

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.

IAM-authorized operaciones de usuario

AdminAddUserToGroup

AdminConfirmSignUp

AdminCreateUser

AdminDeleteSoftwareToken

AdminDeleteUser

AdminDeleteUserAttributes

AdminDisableProviderForUser

AdminDisableUser

AdminEnableUser

AdminForgetDevice

AdminGetDevice

AdminGetUser

AdminGetUserAuthFactors

AdminInitiateAuth

AdminLinkProviderForUser

AdminListDevices

AdminListGroupsForUser

AdminListUserAuthEvents

AdminRemoveUserFromGroup

AdminResetUserPassword

AdminRespondToAuthChallenge

IAM-authorized operaciones de usuario

AdminSetUserMFAPreference

AdminSetUserPassword

AdminSetUserSettings

AdminUpdateAuthEventFeedback

AdminUpdateDeviceStatus

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 este usuario con la operación de API no autenticada. SignUp

Para enviar estas solicitudes en un cliente público que hayas desarrollado con un AWS SDK, no necesitas 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

Operaciones de usuario no autenticadas

ResendConfirmationCode

ForgotPassword

ConfirmForgotPassword

InitiateAuth

Token-authorized operaciones de usuario

Token-authorized las operaciones de usuario cierran sesión, administran las credenciales, modifican y ven a sus usuarios después de que hayan iniciado sesión o hayan iniciado el proceso de inicio de sesión. 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, tu cliente puede incluir el token de acceso del usuario en una solicitud de UpdateUserAttributesAPI.

Para enviar estas solicitudes en un cliente público que hayas desarrollado con un AWS SDK, no necesitas 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>`*.

Token-authorized operaciones de usuario	AccessToken	Session
RespondToAuthChallenge		✓
ChangePassword	✓	
GetUser	✓	
StartWebAuthnRegistration	✓	
CompleteWebAuthnRegistration	✓	
DeleteWebAuthnCredential	✓	
ListWebAuthnCredentials	✓	
UpdateUserAttributes	✓	
DeleteUserAttributes	✓	
DeleteUser	✓	
ConfirmDevice	✓	
ForgetDevice	✓	
GetDevice	✓	
ListDevices	✓	
UpdateDeviceStatus	✓	
GetUserAttributeVerificationCode	✓	

Token-authorized operaciones de usuario	AccessToken	Session
VerifyUserAttribute	✓	
SetUserSettings	✓	
SetUserMFAPreference	✓	
GlobalSignOut	✓	
UpdateAuthEventFeedback		✓
AssociateSoftwareToken	✓	✓
VerifySoftwareToken	✓	✓
RevokeToken[1]		
GetTokensFromRefreshToken[1]		

[1] `RevokeToken` y `GetTokensFromRefreshToken` toman los tokens de actualización como el parámetro de autorización. El token de actualización sirve de token de autorización y de recurso de destino.

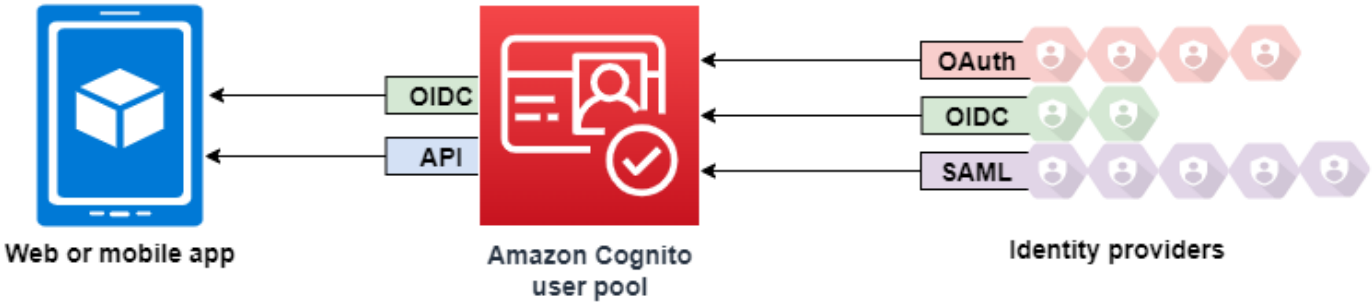
Inicio de sesión en el grupo de usuarios con proveedores de identidad externos

Los usuarios de la aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidades (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, sus sistemas de back-end pueden estandarizarse en un conjunto de tokens de grupo de usuarios.

Cómo funciona el inicio de sesión federado en los grupos de usuarios de Amazon Cognito

Sign-in 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 añade atributos a su usuario en función de las afirmaciones de su IDP y, en el caso del OIDC y de los proveedores de identidad social, de un punto final público. IdP-operated `userinfo` 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 Descripción de los tokens web JSON (JWT) de los 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 las páginas para el inicio de sesión administrado. Para obtener más información, consulte Uso de un dominio propio con el inicio de sesión administrado.
- No puede iniciar sesión con usuarios federados con operaciones de API como y. InitiateAuthAdminInitiateAuth Cuando utiliza los flujos de autenticación integrados de Amazon Cognito, los usuarios federados solo pueden iniciar sesión con el o elPunto de conexión Login. Autorizar punto de conexión Si lo usa`CUSTOM_AUTH`, sus factores desencadenantes controlan el resultado. Son responsables de si el servicio emite un token a cualquier usuario (federado o no).
- 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 el inicio de sesión administrado. De lo contrario, se redirige al Punto de conexión Login del inicio de sesión administrado.

- Cuando el inicio de sesión administrado 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 se ajuste a sus requisitos personalizados, cree una asignación al `preferred_username` atributo. Para obtener más información, consulte Cosas que debe saber acerca de los asignaciones.

  Ejemplo: `MyIDP_bob@example.com`
- Amazon Cognito crea un grupo de usuarios para cada OIDC, SAMI e IdP social que añada a su grupo de usuarios. El nombre del grupo tiene el formato `[user pool ID]_[IdP name]`, por ejemplo, `us-east-1_EXAMPLE_MYSSO` o `us-east-1_EXAMPLE_Google`. Cada perfil de usuario de IdP único generado automáticamente se agrega automáticamente a este grupo. Los usuarios vinculados no se agregan automáticamente a este grupo, pero puede agregar sus perfiles al grupo en un proceso independiente.
- 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 el inicio de sesión administrado.
- Amazon Cognito admite hasta 20 redireccionamientos HTTP entre él y su IdP.
- Cuando el usuario inicia sesión con el inicio de sesión administrado, 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, el inicio de sesión administrado 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 aplicación, y los usuarios solo pueden iniciar sesión con un IdP que hayas asignado a su cliente de aplicación.

Temas

- Configuración de proveedores de identidad para su grupo de usuarios
- Uso de proveedores de identidades de redes sociales con un grupo de usuarios
- Uso de proveedores de identidades SAML con un grupo de usuarios
- Uso de proveedores de identidades de OIDC con un grupo de usuarios
- Asignación de atributos de IdP a perfiles y tokens
- Vinculación de usuarios federados a un perfil de usuario existente

Configuración de proveedores de identidad para su grupo de usuarios

Con los grupos de usuarios, puede implementar el inicio de sesión a través de varios proveedores de identidad externos (IdPs). En esta sección de la guía, encontrará instrucciones para configurar dichos proveedores de identidades con su grupo de usuarios en la consola de Amazon Cognito. Como alternativa, puede usar la API de grupos de usuarios y un AWS SDK para agregar proveedores de identidad de grupos de usuarios mediante programación. Para obtener más información, consulte CreateIdentityProvider.

Las opciones de proveedores de identidades compatibles incluyen proveedores de redes sociales como Facebook, Google o Amazon, así como proveedores OpenID Connect (OIDC) o SAML 2.0. Antes de empezar, configúrese con las credenciales administrativas de su IdP. Deberá registrar su solicitud en cada tipo de proveedor, obtener las credenciales necesarias y luego configurar los detalles del proveedor en su grupo de usuarios. A continuación, los usuarios podrán registrarse e iniciar sesión en la aplicación con las cuentas que poseen en los proveedores de identidades conectados.

El menú de proveedores sociales y externos de Autenticación agrega y actualiza el grupo de usuarios. IdPs Para obtener más información, consulte Inicio de sesión en el grupo de usuarios con proveedores de identidad externos.

En el caso de un IdP de OIDC u OAuth, puede asignar los valores de ACR del IdP a los niveles de autenticación de Amazon Cognito con el campo para permitir la autenticación intensificada para los usuarios federados. `AcrMapping` Para obtener más información, consulte Step-up autenticación para usuarios federados.

Temas

- Configurar el inicio de sesión de los usuarios con un IdP de redes sociales

- Configurar el inicio de sesión de usuarios con un IdP de OIDC
- Configurar el inicio de sesión de usuario con un IdP SAML

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, 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. Seleccione el menú Proveedores sociales y externos.
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 Uso de proveedores de identidades de redes sociales 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`
     - Inicia sesión con Apple: `name email`

   > **Note**
   >
   > Para iniciar sesión con Apple (consola), usa las casillas de verificación para elegir los ámbitos.

7. Seleccione Save changes (Guardar cambios).
8. En el menú Clientes de aplicación, seleccione un cliente de aplicación de la lista y elija Editar. Agregue el nuevo proveedor de identidad social al cliente de aplicación en Identity providers (Proveedores de identidad).
9. Seleccione Save changes (Guardar cambios).

Para obtener más información sobre las redes sociales IdPs, consulta. Uso de proveedores de identidades de redes sociales 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, introduzca sus AWS credenciales.
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. Seleccione el menú Proveedores sociales y externos y, a continuación, seleccione Agregar un proveedor de identidades.
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 los grupos de atributos de usuario (por ejemplo, `name` y`email`) que la aplicación solicita a su 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). Utilice Completar automáticamente a través de la URL del emisor cuando su proveedor tenga un punto de conexión `.well-known/openid-configuration` público desde el que Amazon Cognito pueda recuperar las URL de los puntos de conexión `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 afirmación de la OIDC y seleccione el atributo del 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. En el menú Clientes de aplicación, seleccione un cliente de aplicación de la lista. Para añadir el nuevo proveedor de identidad SAML al cliente de aplicación, vaya a la pestaña Páginas de inicio de sesión y seleccione Editar en Configuración de páginas de inicio de sesión administrado.
17. Seleccione Save changes (Guardar cambios).

Para obtener más información sobre el IdPs OIDC, consulte. Uso de proveedores de identidades de 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. Configuración de un proveedor de identidades de SAML externo

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 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 el menú Proveedores sociales y externos y, a continuación, seleccione Agregar un proveedor de identidades.
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.auth.us-east-1.amazoncognito.com/saml2/logout` que crea Amazon Cognito al configurar el inicio de sesión administrado. 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 IDP de SAML procesa la solicitud de cierre de sesión firmada y cierra 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**
   >
   > 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, por ejemplo`email`, mientras que otros utilizan URL-formatted nombres de atributos similares a los siguientes:

   ```
   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 esta excepción sería «Error al recuperar los metadatos de*<metadata endpoint>*».

Para configurar el proveedor de identidad SAML para añadir un certificado de firma

- Para obtener el certificado que contiene la clave pública que el IdP utiliza para verificar la solicitud de cierre de sesión firmada, haga lo siguiente:
  1. Vaya al menú Proveedores sociales y externos de su grupo de usuarios.
  2. Seleccione su proveedor de SAML.
  3. Elija Ver certificado de firma.

Para obtener más información sobre SAML IdPs , consulte. Uso de proveedores de identidades SAML con un grupo de usuarios

Uso de proveedores de identidades de redes sociales con un grupo de usuarios

Los usuarios de sus aplicaciones web y móviles pueden iniciar sesión a través de proveedores de identidad social (IdP) como Google, Facebook, Amazon y Apple. Activa el inicio de sesión gestionado y Amazon Cognito añade un botón de inicio de sesión para cada IDP social a sus páginas de inicio de sesión gestionado. Amazon Cognito gestiona los intercambios de OAuth 2.0 y OpenID Connect con el proveedor y emite a la aplicación un conjunto estándar de tokens de grupo de usuarios, de modo que sus sistemas de back-end pueden estandarizar los tokens de grupo de usuarios independientemente de cómo haya iniciado sesión el usuario. Para obtener más información sobre los puntos de enlace que Amazon Cognito crea para estos intercambios, consulte la referencia de la API de autenticación de grupos de usuarios de Amazon Cognito.

Cada proveedor tiene su propia consola de desarrollador, sus propios valores y sus propias dificultades. Elija su proveedor para obtener instrucciones paso a paso:

- Añadir a Google como proveedor de identidad social
- Añadir Iniciar sesión con Apple como proveedor de identidad social
- Añadir Facebook como proveedor de identidad social
- Añadir Login with Amazon como proveedor de identidad social

> **Note**
>
> Sign-in 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

- Añadir a Google como proveedor de identidad social
- Añadir Iniciar sesión con Apple como proveedor de identidad social
- Añadir Facebook como proveedor de identidad social
- Añadir Login with Amazon como proveedor de identidad social

Añadir a Google como proveedor de identidad social

> **Note**
>
> Última verificación en la consola del proveedor: 2 de octubre de 2026. Los pasos de la consola de Google siguen los pasos de la plataforma de autenticación de Google (las secciones Marca, Audiencia, Acceso a los datos y Clientes). Si la consola de Google no cumple estos pasos, consulta la documentación de Google Auth Platform.

Para permitir que tus usuarios inicien sesión con Google, registras un cliente de OAuth en la plataforma de autenticación de Google y, a continuación, añades a Google como proveedor de identidad (IdP) en tu grupo de usuarios. Primero debes habilitar el inicio de sesión gestionado.

Registra un cliente OAuth en la plataforma de autenticación de Google

Para registrar un cliente de Google OAuth

1. Inicia sesión en la consola de Google Cloud y crea o selecciona un proyecto.
2. Abre Menu y, a continuación, Google Auth Platform. Si es la primera vez, selecciona Comenzar en la página de información general e introduce el nombre de la aplicación, el correo electrónico de asistencia al usuario, el tipo de audiencia (externa) y la información de contacto.
3. Selecciona Marca. En Dominios autorizados, añade los dominios registrables de tu grupo de usuarios, no las URL completas:
   - `amazoncognito.com`, para el dominio predeterminado de Amazon Cognito.
   - La raíz de su dominio personalizado, por ejemplo`example.com`, si usa uno.
4. Selecciona Acceso a datos y, a continuación, Añadir o eliminar ámbitos. Agregue `openid`, `.../auth/userinfo.email` y `.../auth/userinfo.profile`.
5. Seleccione Clientes y, a continuación, Crear cliente. En Tipo de aplicación, elija Aplicación web.
6. En JavaScript Orígenes autorizados, introduzca el dominio de su grupo de usuarios sin ruta:
   - Dominio predeterminado de Amazon Cognito: `https://<your-prefix>.auth.<region>.amazoncognito.com`
   - Dominio personalizado: `https://auth.example.com`
7. En URIs de redireccionamiento autorizados, introduzca el `/oauth2/idpresponse` punto final del dominio de su grupo de usuarios:

- Dominio predeterminado de Amazon Cognito: `https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse`
- Dominio personalizado: `https://auth.example.com/oauth2/idpresponse`

8. Seleccione Crear. Guarde de forma segura el ID de cliente y el secreto de cliente que muestra Google. Introduce estos valores al añadir Google a su grupo de usuarios.

> **Note**
>
> En el caso de los clientes creados después de junio de 2025, Google muestra el secreto del cliente solo una vez. Guárdalo inmediatamente.

Añade Google a tu grupo de usuarios

Para añadir Google como IdP en la Consola de administración de AWS

1. En la consola de Amazon Cognito, selecciona tu grupo de usuarios, proveedores sociales y externos y, a continuación, Añadir un proveedor de identidades.
2. Seleccione Google. Introduzca el ID de cliente y el secreto del cliente de la plataforma de autenticación de Google.
3. Para los ámbitos autorizados`profile email openid`, ingresa separados por espacios.
4. Asigne los atributos de Google que desee a los atributos de su grupo de usuarios. Como mínimo, mapee `email` hacia `email` y `email_verified` hacia`email_verified`. Si no mapea`email_verified`, Amazon Cognito considerará el correo electrónico como no verificado.
5. Elija Agregar proveedor de identidad y habilite Google en el cliente de su aplicación.

Prueba el inicio de sesión con Google

Abre la página de inicio de sesión gestionado y selecciona Continuar con Google o abre directamente el punto final autorizado:

```
https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=<app-client-id>&redirect_uri=<your-app-
callback>&identity_provider=Google
```

Punto de comprobación

El navegador completa el selector de cuentas de Google y regresa `/oauth2/idpresponse` a tu aplicación con un código de autorización. En la administración de usuarios, aparece un nuevo usuario federado con el proveedor de Google, se rellena `email` y `email_verified` se establece en. `true` No deberías ver ningún `redirect_uri_mismatch` error.

Google-specific trampas

Pantalla de consentimiento no verificado, límite de usuario

Una aplicación con el estado de publicación en fase de prueba se limita a los usuarios de prueba que indiques. Una aplicación que muestre la pantalla de aplicaciones no verificadas tiene un límite de 100 usuarios nuevos durante la duración del proyecto, y ese límite no se puede restablecer. Sin embargo, si tu aplicación solo solicita los ámbitos de inicio de sesión básicos (`openid`,,`profile`)`email`, no es necesario que los usuarios de prueba aparezcan en la lista, no reciban ninguna advertencia y sus autorizaciones no caducan. Respeta estos límites para evitar el límite o verifica tu aplicación en el Centro de verificación.

Redirect-URI propagación

Los cambios en los orígenes autorizados y en las URI de redireccionamiento pueden tardar entre cinco minutos y unas horas en surtir efecto. Vuelve una discrepancia`redirect_uri_mismatch`; vuelve un origen no registrado. `origin_mismatch`

email_verified y vinculación de cuentas

Google hace afirmaciones `email_verified=true` para las cuentas de Gmail y Google Workspace. Amazon Cognito no fusiona automáticamente un usuario de Google con un usuario existente que tenga el mismo correo electrónico. Para vincularlos, implemente la vinculación (por ejemplo, con un activador Lambda previo al registro `AdminLinkProviderForUser` y) y vincule únicamente a un `email_verified` valor confiable.

Añadir Iniciar sesión con Apple como proveedor de identidad social

> **Note**
>
> Última verificación en la consola del proveedor: 2 de octubre de 2026. Apple Steps usa los certificados, identificadores y perfiles en la cuenta de desarrollador de Apple. Debes tener el rol de titular de la cuenta o administrador.

Para iniciar sesión con Apple se necesitan cuatro elementos: un ID de aplicación, un ID de servicios (tu ID de cliente de Amazon Cognito), una clave privada y tu ID de equipo. A continuación, añades Apple como proveedor de identidad (IdP) a tu grupo de usuarios. Primero debes habilitar el inicio de sesión administrado.

Crea tus identificadores y tu clave de Apple

Para crear los artefactos de Apple

1. En Certificados, identificadores y perfiles, crea un ID de aplicación (identificadores, después +, ID de aplicación y, por último, aplicación) y activa la función Iniciar sesión con Apple.
2. Cree un ID de servicios (identificadores, luego + y, por último, ID de servicios). Introduce una descripción y un identificador de dominio inverso, por ejemplo. `com.example.auth.cognito` Este identificador es su ID de Amazon Cognito Services (ID de cliente).
3. Seleccione el ID de servicios, active Iniciar sesión con Apple y elija Configurar. Elige tu ID de aplicación principal y, a continuación, introduce las URL web:

   - Dominios y subdominios (solo host, sin esquema): `<your-prefix>.auth.<region>.amazoncognito.com` para el dominio predeterminado de Amazon Cognito `auth.example.com` o para un dominio personalizado.
   - URL devueltas (URL completa con`https://`): `https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse` o `https://auth.example.com/oauth2/idpresponse` para un dominio personalizado.
4. Crea una clave privada (claves y, luego, +), activa Iniciar sesión con Apple y descarga el `.p8` archivo. Puedes descargarlo solo una vez. Registre la ID clave.
5. Registra tu ID de equipo de 10 caracteres en la página de membresía. Introduce el ID de servicios, el ID del equipo, el ID de clave y la clave privada al añadir Apple a tu grupo de usuarios.

Agrega Apple a tu grupo de usuarios

Para añadir Apple como IdP en la Consola de administración de AWS

1. En la consola de Amazon Cognito, selecciona tu grupo de usuarios, proveedores sociales y externos, añade un proveedor de identidad e inicia sesión con Apple.
2. Introduzca el ID de servicios, el ID del equipo, el ID de clave y la clave privada. Amazon Cognito usa la clave privada para generar el secreto de cliente por usted.

3. Para los ámbitos autorizados, introduzca. `email name` Úselo solo `email` si no necesita el nombre del usuario.
4. `email`Asigne `email` a y `sub` hacia su atributo de nombre de usuario. Apple devuelve el nombre del usuario solo la primera vez que inicia sesión, así que captúralo si lo mapeas.
5. Selecciona Añadir proveedor de identidad y activa Apple en el cliente de tu aplicación.

Prueba Iniciar sesión con Apple

Abre la página de inicio de sesión gestionado y selecciona Continuar con Apple. Prueba Compartir mi correo electrónico y Ocultar mi correo electrónico.

Punto de comprobación

El navegador vuelve `/oauth2/idpresponse` a tu aplicación con un código de autorización. Aparece un usuario en tu grupo de usuarios con el SignInWithAppleproveedor y un`sub`. La ejecución Compartir mi correo electrónico muestra el correo electrónico real; la ejecución Ocultar mi correo electrónico muestra una `@privaterelay.appleid.com` dirección. Un `invalid_client` error casi siempre significa que el JWT secreto del cliente ha caducado o que el identificador del servicio o la URL de retorno no coinciden exactamente. `/oauth2/idpresponse`

Apple-specific trampas

El secreto del cliente es un JWT que caduca

El secreto de cliente de Apple no es una cadena estática. Es un ES256-signed JWT con una vida útil máxima de seis meses (15 777 000 segundos) y, cuando caduca, el inicio de sesión en Apple se interrumpe sin `invalid_client` previo aviso. En la configuración que se describe aquí, proporciona a Amazon Cognito la clave privada (junto con el ID de servicios, el ID del equipo y el ID de clave) y Amazon Cognito genera el JWT secreto del cliente a partir de esa clave; no se pega un secreto prediseñado. Mantén la clave privada que registraste en Apple como válida y actualizada en la configuración del IdP; si revocas o sustituyes la clave de Apple, actualiza el iDP (por ejemplo, con`UpdateIdentityProvider`) con la nueva clave.

Retransmisión privada de correo electrónico y vinculación de cuentas

Cuando un usuario elige Ocultar mi correo electrónico, Apple devuelve una `@privaterelay.appleid.com` dirección en lugar de la verdadera. Esta dirección es diferente del correo electrónico que la misma persona usa con otros proveedores, así que no vincules cuentas de proveedores de identidad por correo electrónico cuando Apple esté implicado, sino

que vincules por proveedor. `sub` Para enviar correo a una dirección de retransmisión, registra tus dominios de envío en Iniciar sesión con Apple para comunicarte por correo electrónico.

tipo email_verified y cuentas administradas

La `email_verified` afirmación de Apple puede llegar en forma de cadena `"true"` o `true` booleana (maneje ambos). Siempre es cierto para las cuentas de Apple de consumo. En el caso de las cuentas (gestionadas) de Apple at Work & School, puede ser falsa. Prefiera la vinculación `sub` basada en bases y normalice el valor de la cadena antes que cualquier lógica de fusión.

Añadir Facebook como proveedor de identidad social

> **Note**
>
> Última verificación en la consola del proveedor: 2 de octubre de 2026. Los pasos de Facebook utilizan la consola Meta for Developers, que organiza la configuración de las aplicaciones en función de los casos de uso. La configuración de redireccionamiento de OAuth se encuentra en Casos de uso, Personalización y Configuración.

Para permitir que tus usuarios inicien sesión con Facebook, debes crear una aplicación Meta con el caso de uso de inicio de sesión con Facebook y, a continuación, añadir Facebook como proveedor de identidad (IdP) a tu grupo de usuarios. Primero debes habilitar el inicio de sesión administrado.

Cree una aplicación Meta para iniciar sesión en Facebook

Para crear y configurar una aplicación Meta

1. Inicia sesión en Meta for Developers, selecciona Mis aplicaciones y, a continuación, Crear aplicación.
2. Para este caso de uso, selecciona Autenticar y solicita datos a los usuarios con el inicio de sesión en Facebook. Termina de crear la aplicación. Las aplicaciones nuevas se inician en el modo de desarrollo.
3. En los ajustes de la aplicación y, a continuación, en Básico, registra el ID y el secreto de la aplicación. Configura App Domains en el host (`<your-prefix>.auth.<region>.amazoncognito.comoauth.example.com`) del dominio de tu grupo de usuarios y añade la plataforma del sitio web con la URL de inicio de sesión gestionado como URL del sitio.

4. Ve a Casos de uso, selecciona tu caso de uso de inicio de sesión con Facebook y, a continuación, Personalizar. Confirma el `public_profile` permiso (obligatorio) y añádeloemail.
5. En la configuración de casos de uso, establece los URI de redireccionamiento de OAuth válidos en el `/oauth2/idpresponse` punto final del dominio de tu grupo de usuarios:
   - Dominio predeterminado de Amazon Cognito: `https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse`
   - Dominio personalizado: `https://auth.example.com/oauth2/idpresponse`

Añada Facebook a su grupo de usuarios

Para añadir Facebook como IdP en la Consola de administración de AWS

1. En la consola de Amazon Cognito, selecciona tu grupo de usuarios, después proveedores sociales y externos, Añadir un proveedor de identidad y, por último, Facebook.
2. Introduzca el ID de la aplicación y el secreto de la aplicación y elija una versión de la API de Facebook. Elige la última versión disponible, ya que cada versión tiene una fecha de descontinuación y su alcance puede variar de una versión a otra.
3. Para los ámbitos autorizados, introduzca los campos separados por `public_profile,email` comas.
4. Asigne un mapa `email` hacia `email` y hacia. `name` `name`
5. Selecciona Añadir proveedor de identidad y activa Facebook en el cliente de tu aplicación.

Prueba el inicio de sesión en Facebook

Mientras la aplicación esté en modo de desarrollo, añade tu propia cuenta de Facebook como rol de aplicación (administrador, desarrollador o probador). A continuación, abre la página de inicio de sesión gestionado y selecciona Continuar con Facebook.

Punto de comprobación

Facebook vuelve a acceder `/oauth2/idpresponse` a tu aplicación con un código de autorización y aparece un nuevo usuario federado en tu grupo de usuarios con `email` un texto rellenado. `name` A continuación, cambia la aplicación al modo en vivo y confirma que una cuenta que no sea de rol pueda iniciar sesión. Una cuenta sin rol que muestre «la aplicación no está activa» o «en modo de

desarrollo» significa que la aplicación aún se encuentra en modo de desarrollo o carece de acceso avanzado.

Facebook-specific dificultades

Modo de desarrollo y revisión de la aplicación

Una nueva aplicación está en modo de desarrollo y solo los usuarios con funciones de aplicación (administrador, desarrollador, probador) pueden iniciar sesión. Para permitir que el público en general inicie sesión, cambie la aplicación al modo en vivo, complete la verificación empresarial y obtenga acceso avanzado `email` (y mayor acceso para). `public_profile` First-time Los evaluadores suelen pulsar «la aplicación no está activa» porque realizan las pruebas con una cuenta que no es de rol.

No hay ninguna reclamación verificada por email_verified

Facebook no devuelve ninguna `email_verified` reclamación, por lo que no hay nada que mapear, y Amazon Cognito considera que el correo electrónico de Facebook no está verificado. No dé por sentado que se trata de un correo electrónico verificado de Facebook y vincule las cuentas por correo electrónico únicamente mediante una lógica deliberada y fiable, en lugar de hacerlo de forma automática.

Mantén los alcances pequeños

Sign-in la finalización se reduce drásticamente cuando se superan los cuatro permisos. Mantenga la solicitud en `public_profile` y`email`. La federación de Amazon Cognito tampoco es compatible con el inicio de sesión limitado de Facebook en iOS.

Añadir Login with Amazon como proveedor de identidad social

> **Note**
>
> Última verificación en la consola del proveedor: 2 de octubre de 2026. Los pasos de Login with Amazon utilizan la consola Amazon Developer (no la AWS consola).

Para permitir que tus usuarios inicien sesión con su cuenta de Amazon, registras un perfil de seguridad de Login with Amazon y, a continuación, añades Login with Amazon como proveedor de identidad (IdP) en tu grupo de usuarios. Primero debes habilitar el inicio de sesión gestionado.

Registrar un perfil de seguridad de Login with Amazon

Para crear y configurar un perfil de seguridad

1. Inicia sesión en la consola de desarrolladores de Amazon o ve a Aplicaciones y servicios y, a continuación, a Login with Amazon.
2. Selecciona Crear un nuevo perfil de seguridad. Introduzca un nombre (que aparece en la pantalla de consentimiento), una descripción y la URL del aviso de privacidad del consentimiento. Seleccione Save.
3. Revela el ID de cliente y el secreto del cliente para el perfil de seguridad. Los introduce al añadir Login with Amazon a su grupo de usuarios.
4. Abre la configuración web del perfil y selecciona Editar. Introduce los valores que aparecen a continuación y, a continuación, guarda.

   - Orígenes permitidos (HTTPS, sin ruta): `https://<your-prefix>.auth.<region>.amazoncognito.com` o`https://auth.example.com`.
   - URL de retorno permitidas (el `/oauth2/idpresponse` punto final): `https://<your-prefix>.auth.<region>.amazoncognito.com/oauth2/idpresponse` o`https://auth.example.com/oauth2/idpresponse`. Amazon Cognito usa el flujo de redireccionamiento, por lo que la URL de retorno es el valor del que depende.

Añade Login with Amazon a tu grupo de usuarios

Para añadir Login with Amazon como IdP en la Consola de administración de AWS

1. En la consola de Amazon Cognito, selecciona tu grupo de usuarios, proveedores sociales y externos, añade un proveedor de identidad y, por último, Login with Amazon.
2. Introduzca el ID de cliente y el secreto del cliente del perfil de seguridad.
3. Para los ámbitos autorizados, introduzca`profile`. Esto devuelve el nombre, el correo electrónico y el seudónimo del usuario. (`postal_code`Añádelo solo si necesitas el código postal o postal).
4. Asigne `email` a `emailname`, `name` a y `user_id` al atributo de su nombre de usuario.
5. Selecciona Añadir proveedor de identidad y activa Login with Amazon en el cliente de tu aplicación.

Prueba Login with Amazon sign-in

Abre la página de inicio de sesión gestionado y selecciona Continuar con Amazon y, a continuación, autentica y acepta la pantalla de consentimiento.

Punto de comprobación

El navegador vuelve `/oauth2/idpresponse` a tu aplicación con un código de autorización y el usuario aparece en tu grupo de usuarios como un usuario federado cuyo nombre de usuario lleva un prefijo. `LoginWithAmazon_` El token de ID contiene el y asignado`email`. `name` Tenga en cuenta que `email_verified` se lee `false` a menos que lo configure con un disparador (consulte las dificultades).

Login with Amazon pitfalls

No hay ninguna reclamación verificada por correo electrónico

El `profile` alcance solo se devuelve y `email`name, no `user_id` hay ninguna `email_verified` pretensión de mapear. Los correos electrónicos mapeados no están verificados de forma predeterminada, por lo que los usuarios de Login with Amazon aparecen con el valor `email_verified` establecido en. `false`

Efecto en la vinculación de cuentas

Como el correo electrónico no está verificado, Amazon Cognito no fusionará automáticamente un usuario de Login with Amazon con un usuario verificado por correo electrónico existente. Para vincular o marcar el correo electrónico como verificado, utilice un activador Lambda previo al registro (`AdminLinkProviderForUsery`) y solo cuando su modelo de confianza justifique la confianza en el correo electrónico. Amazon-asserted

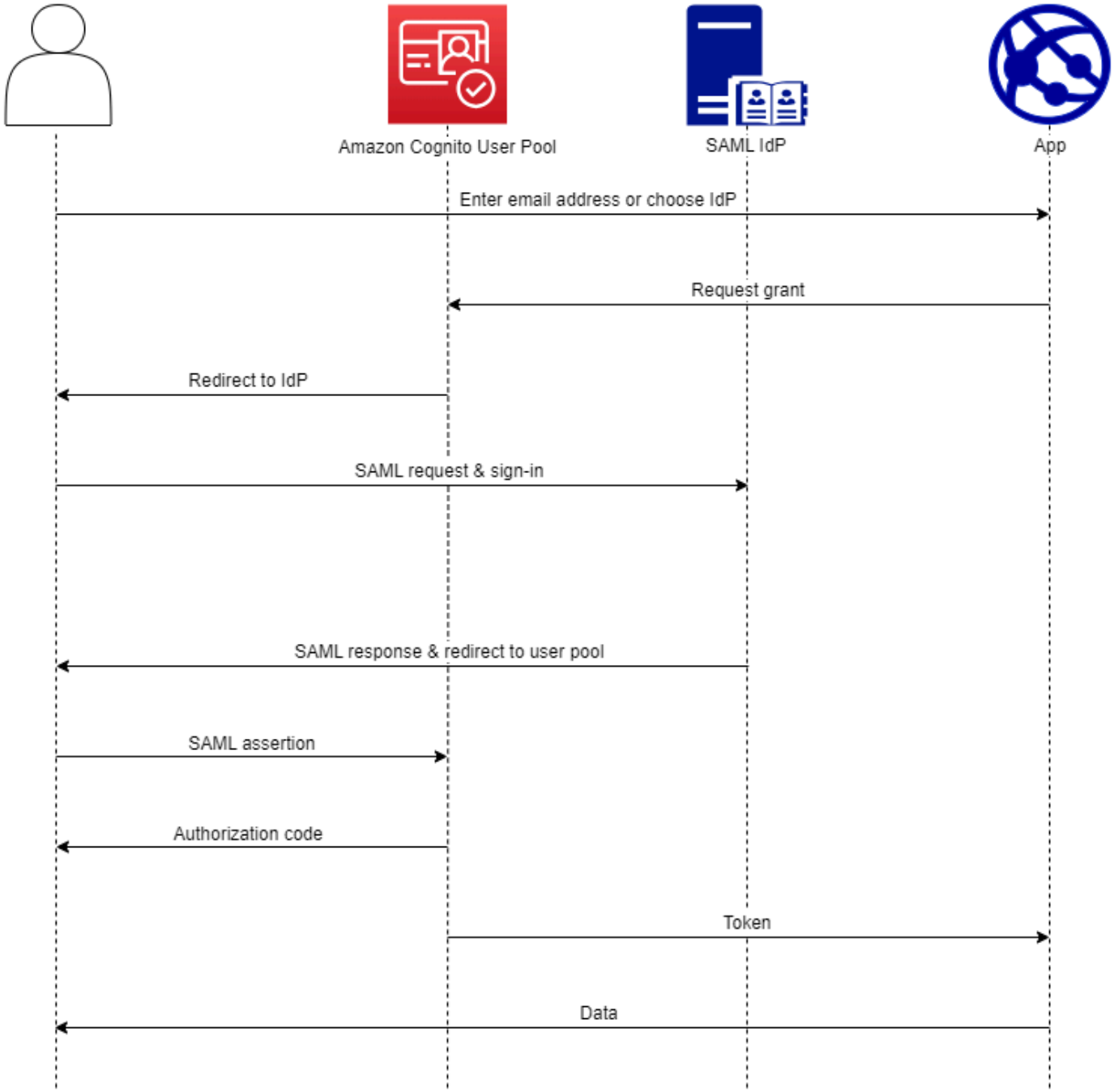
Los atributos mapeados deben poder escribirse

El cliente de la aplicación debe poder mutar y escribir cualquier atributo personalizado que asigne; de lo contrario, al iniciar sesión se producirá un error o se omitirá el atributo de forma automática. Estándar y están bien. `email` `name`

Uso de proveedores de identidades SAML con un grupo de usuarios

Puede elegir que los usuarios web y de aplicaciones móviles inicien sesión a través de un proveedor de identidades (IdP) SAML como Microsoft Active Directory Federation Services (ADFS) o Shibboleth. Debe elegir un IdP SAML compatible con el estándar SAML 2.0.

Con el inicio de sesión administrado, Amazon Cognito autentica usuarios de IdP locales y de terceros y emite tokens web JSON (JWT). Con los tokens que emite Amazon Cognito, puede consolidar varios orígenes de identidad en un estándar universal de OpenID Connect (OIDC) en todas sus aplicaciones. Amazon Cognito puede procesar las aserciones de SAML de los proveedores externos y convertirlas en ese estándar de SSO. Puede crear y administrar un IDP de SAML en Consola de administración de AWS, a través de o con AWS CLI la API de grupos de usuarios de Amazon Cognito. Para crear su primer IdP de SAML en Consola de administración de AWS, consulte. Adición y administración de proveedores de identidad de SAML a 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 el IdP SAML para que acepte solicitudes y envíe respuestas a su grupo de usuarios. La documentación de su IdP de SAML contiene información sobre cómo añadir su grupo de usuarios como parte de confianza o aplicación para su IdP de SAML 2.0. En la documentación siguiente se indican los valores que debe proporcionar para el ID de entidad del SP y la URL de servicio de consumidor de aserción (ACS).

Referencia rápida de los valores de SAML del grupo de usuarios

ID de entidad del SP

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

URL del ACS

```
https://Your user pool domain/saml2/idpresponse
```

Debe configurar el grupo de usuarios para que sea compatible con el proveedor de identidades. A continuación, indicamos los pasos generales para agregar un IdP SAML externo.

1. Descargue los metadatos de SAML de su IdP o recupere la URL del punto de conexión de metadatos. Consulte Configuración de un proveedor de identidades de SAML externo.
2. Añada el IdP nuevo al grupo de usuarios. Cargue los metadatos de SAML o proporcione la URL de los metadatos. Consulte Adición y administración de proveedores de identidad de SAML a un grupo de usuarios.
3. Asigne el IdP a los clientes de aplicación. Consulte Application-specific configuración con clientes de aplicaciones.

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
- Configuración de un proveedor de identidades de SAML externo

- Adición y administración de proveedores de identidad de SAML a un grupo de usuarios
- Inicio de sesión SAML en grupos de usuarios de Amazon Cognito
- Cierre de sesión de usuarios de SAML con un cierre de sesión único
- Firma y cifrado de SAML
- Nombres e identificadores de proveedor de identidades SAML

Información que debe saber sobre SAML IdPs en los grupos de usuarios de Amazon Cognito

La implementación de un IdP SAML 2.0 está sujeta a algunos requisitos y restricciones. Consulte esta sección cuando implemente el IdP. También encontrará información útil para solucionar errores durante la federación de SAML con un grupo de usuarios.

Amazon Cognito procesa las aserciones SAML en su lugar

> Los grupos de usuarios de Amazon Cognito admiten la federación de SAML 2.0 con puntos de enlace. 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 IdP-initiated único (SSO), tal SP-initiated y como se describe en las secciones 5.1.2 y 5.1.4 de la descripción técnica de SAML. V2.0

Facilitación de un certificado de firma de IdP válido

> El certificado de firma de los metadatos del proveedor SAML no debe estar caducado cuando configure el IdP 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 de longitud.

Conservación del 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 token `relayState` es una referencia opaca a la información de estado que Amazon Cognito mantiene. 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.

Identificación del punto de conexión 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.auth.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.

Imposibilidad de repetir aserciones reproducidas

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 entidad del SP

Debe proporcionar el IdP con el ID del grupo de usuarios en el `urn` del proveedor de servicios (SP), también denominado URI de audiencia o ID de entidad del SP. El URI de destino del grupo de usuarios tiene el siguiente formato.

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

Puede encontrar el ID del grupo de usuarios en la opción Información general sobre el grupo de usuarios de la consola de Amazon Cognito.

Asignación de todos los atributos obligatorios

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 Asignación de atributos de IdP a perfiles y tokens.

El formato de aserción tiene requisitos específicos

El IdP SAML debe incluir las siguientes notificaciones en la aserción SAML.

- Una notificación `NameID`. Amazon Cognito asocia una aserción SAML al usuario de destino mediante `NameID`. Si `NameID` cambia, Amazon Cognito considerará que la afirmación es para un usuario nuevo. El atributo que defina en `NameID` en la configuración del IdP debe tener un valor persistente. Para asignar usuarios de SAML a un perfil de usuario coherente en el grupo de usuarios, asigne a la notificación `NameID` de un atributo un valor que no cambie.

```
<saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:persistent">
  carlos
</saml2:NameID>
```

  Un valor `Format` en la notificación `NameID` del IdP de `urn:oasis:names:tc:SAML:1.1:nameid-format:persistent` indica que el IdP está transfiriendo un valor inmutable. Amazon Cognito no necesita esta declaración de formato y asigna el formato `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified` si el IdP no especifica un formato para la notificación `NameID`. Este comportamiento cumple la sección 2.2.2 Complex Type NameIDType de la especificación de SAML 2.0.

- Una notificación `AudienceRestriction` con un valor `Audience` que establezca el ID de entidad del SP del grupo de usuarios como objetivo de la respuesta.

```
<saml:AudienceRestriction>
  <saml:Audience> urn:amazon:cognito:sp:us-east-1_EXAMPLE
</saml:AudienceRestriction>
```

- Para el inicio de sesión SP-initiated único, un `Response` elemento con el valor del ID de solicitud de SAML original. `InResponseTo`

```
<saml2p:Response Destination="https://mydomain.auth.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**
>
> IdP-initiated Las aserciones de SAML no deben contener ningún valor. `InResponseTo`

- Un `SubjectConfirmationData` elemento con un `Recipient` valor del `saml2/idpresponse` punto final del grupo de usuarios y, en el caso de SP-initiated SAML, un `InResponseTo` valor que coincida con el identificador de solicitud de SAML original.

```
<saml2:SubjectConfirmationData InResponseTo="_dd0a3436-bc64-4679-a0c2-
cb4454f04184" NotOnOrAfter="Date-time stamp" Recipient="https://mydomain.auth.us-
east-1.amazoncognito.com/saml2/idpresponse"/>
```

SP-initiated solicitudes de inicio de sesión

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 el punto de conexión del ACS. Si lo desea, puede aplicar una firma criptográfica a estas solicitudes.

Firma de solicitudes y cifrado de las respuestas

Cada grupo de usuarios con un proveedor SAML genera un par de claves asimétricas y un certificado de firma para una firma digital que Amazon Cognito asigna a las solicitudes de SAML. Cada IdP SAML externo que configure para admitir una respuesta de SAML cifrada hace que Amazon Cognito genere un nuevo par de claves y un certificado de cifrado para dicho proveedor. Para ver y descargar los certificados con la clave pública, seleccione su IdP en el menú Proveedores sociales y externos de la consola de Amazon Cognito.

Para establecer confianza con las solicitudes de SAML de su grupo de usuarios, proporcione al IdP una copia del certificado de firma de SAML 2.0 del grupo de usuarios. El IdP podría no tener en cuenta las solicitudes de SAML firmadas por su grupo de usuarios si no configura el IdP para que acepte las solicitudes 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) de cualquier IdP externo de SAML. Cuando proporciona una copia del certificado, el IdP puede comprobar la integridad de las solicitudes de SAML de los usuarios.
2. Su IdP SAML puede cifrar las respuestas de SAML con el certificado de cifrado. Cuando configure un IdP con cifrado de SAML, su IdP solo debe enviar respuestas cifradas.

Codificación de caracteres no alfanuméricos

Amazon Cognito no acepta UTF-8 caracteres de 4 bytes como los 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 acepta la afirmación 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>
```

A diferencia del ejemplo anterior, se acepta la siguiente afirmación 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">8J+YkA==</saml2:AttributeValue>
</saml2:Attribute>
```

El punto de conexión de los metadatos debe tener una seguridad de capa de transporte válida

Si `InvalidParameterException` al crear un IDP de SAML con una URL de punto de enlace de metadatos HTTPS, por ejemplo, ve «Error al recuperar los metadatos *<metadata endpoint>* de», asegúrese de que el punto de enlace de metadatos tenga SSL correctamente configurado y de que haya un certificado SSL válido asociado a él. Para obtener más información sobre la validación de certificados, consulta ¿Qué es un certificado? SSL/TLS .

El punto de conexión de metadatos debe estar en un puerto TCP estándar para HTTP o HTTPS

Amazon Cognito solo acepta direcciones URL de metadatos para proveedores de SAML en los puertos TCP estándar 80 para HTTP y 443 para HTTPS. Como práctica recomendada de

seguridad, aloje los metadatos de SAML en una TLS-encrypted URL con el `https://` prefijo. Introduzca las direcciones URL de los metadatos en el formato *`http://www.example.com/saml2/metadata.xml`* o *`https://www.example.com/saml2/metadata.xml`*. La consola de Amazon Cognito solo acepta direcciones URL de metadatos con el prefijo `https://`. También puede configurar los metadatos del IdP con CreateIdentityProvidery. UpdateIdentityProvider

Los clientes de aplicaciones con IdP-initiated SAML solo pueden iniciar sesión con SAML

Cuando activas la compatibilidad con un IDP de SAML 2.0 que IdP-initiated admite el inicio de sesión en un cliente de aplicaciones, solo puedes añadir otro SAML IdPs 2.0 a ese cliente de aplicación. No podrá añadir el directorio de usuarios en el grupo de usuarios ni tampoco todos los proveedores de identidades externos que no sean de SAML a un cliente de aplicación configurado de esta manera.

Las respuestas de cierre de sesión deben utilizar enlace POST

El punto de conexión `/saml2/logout` acepta `LogoutResponse` como solicitudes de `HTTP POST`. Los grupos de usuarios no aceptan respuestas de cierre de sesión con enlace `HTTP GET`.

Rotación de certificados de firma de metadatos

Amazon Cognito almacena en caché los metadatos de SAML durante un máximo de seis horas cuando proporciona los metadatos con una URL. Al realizar cualquier rotación de certificados de firma de metadatos, configure su fuente de metadatos para publicar tanto el certificado original como el nuevo durante al menos seis horas. Cuando Amazon Cognito actualiza la caché desde la URL de los metadatos, considera que cada certificado es válido, y su IdP de SAML puede empezar a firmar las aserciones de SAML con el nuevo certificado. Una vez transcurrido este período, puede eliminar el certificado original de los metadatos publicados.

Distinción entre mayúsculas y minúsculas en los nombres de usuario SAML

Cuando un usuario federado intenta iniciar sesión, el proveedor de identidades (IdP) SAML pasa un `NameId` único a Amazon Cognito en la aserción SAML del usuario. Amazon Cognito identifica a un SAML-federated usuario mediante su afirmación. `NameId` Independientemente de la configuración de distinción entre mayúsculas y minúsculas del grupo de usuarios, Amazon Cognito reconoce un usuario federado que regresa de un IdP SAML cuando pasa su notificación `NameId` única que distingue entre mayúsculas y minúsculas. 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 sus usuarios no pueden iniciar sesión después de `NameID` realizar los cambios, elimine sus perfiles de usuario del grupo de usuarios. Amazon Cognito crea nuevos perfiles de usuario la próxima vez que inicien sesión.

Temas

- Configuración de un proveedor de identidades de SAML externo
- Adición y administración de proveedores de identidad de SAML a un grupo de usuarios
- Inicio de sesión SAML en grupos de usuarios de Amazon Cognito
- Cierre de sesión de usuarios de SAML con un cierre de sesión único
- Firma y cifrado de SAML
- Nombres e identificadores de proveedor de identidades SAML

Configuración de un proveedor de identidades de SAML externo

Si desea agregar un proveedor de identidades (IdP) de SAML a su grupo de usuarios, debe introducir algunas actualizaciones de configuración en la interfaz de administración de su IdP. En esta sección se describe cómo dar formato a los valores que debe proporcionar a su IdP. También puede obtener información sobre cómo recuperar el documento de metadatos de URL estática o activa que identifica el IdP y las notificaciones de SAML en el grupo de usuarios.

Para configurar las soluciones del proveedor de identidades (IdP) SAML 2.0 de terceros para que funcionen con la federación de grupos de usuarios de Amazon Cognito, debe configurar el IdP SAML para que realice una redirección a la siguiente URL del servicio de consumidor de aserción (ACS): `https://`*`mydomain.auth.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 el menú Dominio de su grupo de usuarios en la consola de Amazon Cognito.

Algunos tipos de SAML IdPs requieren que introduzca en el `urn` formulario el identificador URI de audiencia o identificador de entidad del SP. `urn:amazon:cognito:sp:`*`us-east-1_EXAMPLE`* Puede encontrar el ID del grupo de usuarios en la opción Información general sobre el grupo de usuarios de la consola de Amazon Cognito.

Asimismo, debe configurar el IdP SAML para que proporcione los valores de todos los atributos designados como atributos obligatorios en el grupo de usuarios. Normalmente, `email` es un atributo obligatorio para los grupos de usuarios, y en tal caso, el IdP SAML deberá proporcionar algún tipo de notificación `email` en su aserción SAML y será preciso asignar la notificación del atributo de dicho 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 el IdP para que confíe en las solicitudes firmadas por el certificado de firma del grupo de usuarios. Para aceptar respuestas de SAML cifradas, debe configurar el IdP para que cifre todas las respuestas de SAML del grupo de usuarios. El proveedor tiene documentación sobre la configuración de estas funciones. Para ver un ejemplo de Microsoft, consulte Configuración del cifrado de tokens SAML de Microsoft Entra.

> **Note**
>
> Amazon Cognito solo necesita el documento de metadatos del proveedor de identidades. Es posible que su proveedor también ofrezca información de configuración personalizada para la federación de SAML 2.0 con IAM o AWS IAM Identity Center. Para obtener información sobre cómo configurar la integración de Amazon Cognito, consulte las instrucciones generales para recuperar el documento de metadatos y administrar el resto de la configuración de su grupo de usuarios.

Solución	Más información
ID de Microsoft Entra	Metadatos de federación
Okta	How to Download the IdP Metadata and SAML Signing Certificates for a SAML App Integration
Auth0	Configure Auth0 as SAML Identity Provider
Identidad de ping (PingFederate)	Exportación de metadatos de SAML desde PingFederate
JumpCloud	SAML Configuration Notes

Solución	Más información
SecureAuth	SAML application integration

Adición y administración de proveedores de identidad de SAML a un grupo de usuarios

Tras configurar el proveedor de identidades para que funcione con Amazon Cognito, podrá añadirlo a los grupos de usuarios y los clientes de aplicación. En los siguientes procedimientos se muestra cómo crear, modificar y eliminar proveedores de SAML en un grupo de usuarios de Amazon Cognito.

Consola de administración de AWS

Puede usarlo Consola de administración de AWS para crear y eliminar proveedores de identidad de SAML (). IdPs

Para poder crear un IdP SAML, necesitará el documento de metadatos de SAML facilitado por el IdP externo. Para obtener instrucciones sobre cómo obtener o generar el documento de metadatos de SAML necesario, consulte Configuración de un proveedor de identidades de 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. Seleccione el menú Proveedores sociales y externos y, a continuación, seleccione Agregar un proveedor de identidades.
5. Elija un IdP SAML.
6. Introduzca un Nombre del proveedor. Puede pasar este nombre descriptivo en un parámetro de solicitud `identity_provider` al Autorizar 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.auth.us-east-1.amazoncognito.com/saml2/`

`logout` que se crea al configurar el inicio de sesión administrado. El punto de conexión `saml2/logout` utiliza el enlace POST.

> **Note**
>
> Si se selecciona esta opción y el IdP SAML espera una solicitud de cierre de sesión firmada, también debe proporcionar al IdP SAML el certificado de firma del grupo de usuarios.
> El IDP de SAML procesa la solicitud de cierre de sesión firmada y cierra la sesión del usuario de Amazon Cognito.

9. Elija su IdP-initiated configuración de inicio de sesión con SAML. Como práctica recomendada de seguridad, selecciona Aceptar solo las aserciones de SP-initiated SAML. Si ha preparado su entorno para aceptar de forma segura las sesiones de inicio de sesión de SAML no solicitadas, elija Aceptar y afirmaciones de SAML. SP-initiated IdP-initiated 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**
    >
    > Le recomendamos que introduzca la URL de un documento de metadatos si su proveedor dispone de un punto de conexión público en lugar de cargar un archivo. Amazon Cognito actualiza automáticamente los metadatos desde la URL de metadatos. Normalmente, los metadatos se actualizan cada seis horas o antes de que caduquen, lo que ocurra primero.

11. Asigne atributos entre el proveedor SAML y el grupo de usuarios para asignar atributos de proveedor SAML al perfil de usuario de 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.

Algunas IdPs usan nombres simples, como, mientras que otras usan nombres como `email` 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 IdP-to-SP respuesta.

  ```
  https://mydomain.auth.us-east-1.amazoncognito.com/saml2/idpresponse
  ```

- El IdP SAML debe rellenar `NameID` y todos los atributos obligatorios para el grupo de usuarios en la aserción SAML. `NameID` se utiliza para identificar al usuario federado de SAML de forma inequívoca en el grupo de usuarios. El IdP debe pasar el ID de nombre SAML de cada usuario en

un formato coherente y que distinga mayúsculas de 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 el IdP pueda utilizar para validar las solicitudes de cierre de sesión de SAML, elija el menú Proveedores sociales y externos del grupo de usuarios, seleccione el 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 el menú Proveedores sociales y externos.
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) y el SSO iniciados por el proveedor de servicios (SP-initiated). IdP-initiated Como práctica de seguridad recomendada, implemente el inicio de sesión único en su grupo de usuarios. SP-initiated En la sección 5.1.2 de la descripción V2.0 técnica del SAML se describe el SSO. SP-initiated 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 SP-initiated flujo, siempre deben realizar primero una solicitud a Amazon Cognito y redirigirlos al IdP para la 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 IdP-initiated que sea compatible con el SSO. Cuando admite la IdP-initiated autenticación, 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 SP-initiated SSO, Amazon Cognito establece parámetros de estado que validan una respuesta de SAML con respecto a la solicitud original. Con SP-initiated el inicio de sesión, también puede protegerse contra la falsificación de solicitudes entre sitios (CSRF).

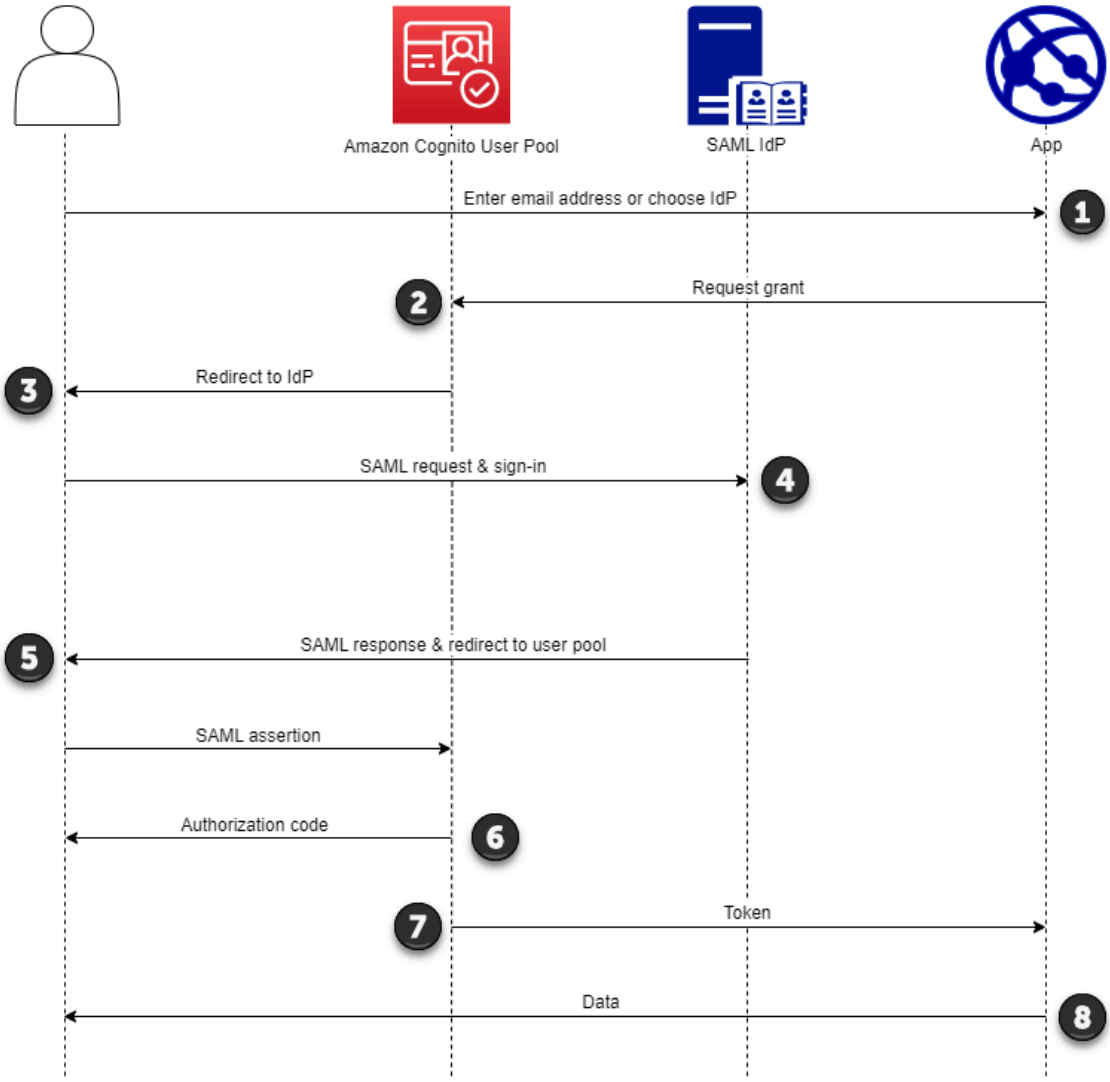
Temas

- Implementa el SP-initated inicio de sesión mediante SAML
- Implemente el inicio de IdP-initiated sesión mediante SAML

Implementa el SP-initated inicio de sesión mediante SAML

Como práctica recomendada, implemente el inicio de sesión iniciado por el proveedor de servicios (SP-initiated) en su grupo de usuarios. Amazon Cognito inicia la sesión del usuario y lo redirige al IdP. Con este método, tiene un amplio control sobre quién presenta las solicitudes de inicio de sesión. También puede permitir IdP-initiated el inicio de sesión en determinadas condiciones.

El siguiente proceso muestra cómo los usuarios completan el inicio de SP-initiated sesión en tu grupo de usuarios a través de un proveedor de SAML.



1. El usuario introduce su dirección de correo electrónico en la página de inicio de sesión. Para determinar la redirección del usuario a su IdP, puede recopilar su dirección de correo electrónico en una aplicación personalizada o invocar el inicio de sesión administrado la vista web.

   Puedes configurar tus páginas de inicio de sesión gestionadas para que muestren una lista IdPs o soliciten una dirección de correo electrónico y la asocien con el identificador de tu IDP de SAML. Para solicitar una dirección de correo electrónico, edite el estilo de marca de su inicio de sesión administrado. En Base, busque Comportamiento de autenticación y, en Pantalla de proveedor, ponga Estilo de visualización como Entrada de búsqueda de dominio.
2. La aplicación invoca el punto de conexión de redireccionamiento del grupo de usuarios y solicita una sesión con el ID de cliente que corresponde a la aplicación y el ID del 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 cinco minutos y redirige al usuario al inicio de sesión administrado. Cuando el usuario obtenga este resultado, recibirá un mensaje de error `Something went wrong`.

6. Tras verificar la aserción SAML y asignar los atributos del usuario desde las notificaciones de la respuesta, Amazon Cognito crea o actualiza internamente el perfil del usuario en el grupo de usuarios. Por lo general, el grupo de usuarios devuelve un código de autorización a la sesión del navegador del usuario.
7. El usuario presenta su código de autorización a la aplicación, que lo intercambia por tokens web JSON (JWT).
8. La aplicación acepta y procesa el token de identificación del usuario como autenticación, genera solicitudes autorizadas a los recursos con un token de acceso y almacena el token de actualización.

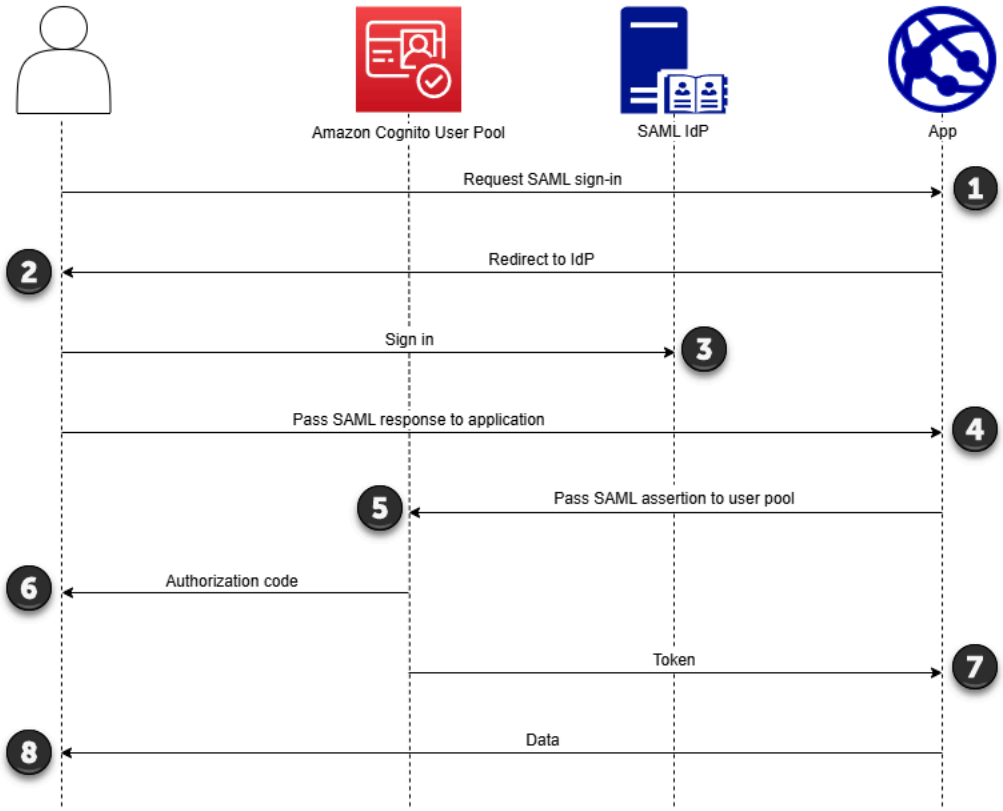
Cuando un usuario se autentica y recibe una adjudicación de código de autorización, el grupo de usuarios devuelve tokens de ID, de acceso y de actualización. El token de ID es un objeto de autenticación para la administración de identidades. OIDC-based El token de acceso es un objeto de autorización con ámbitos de OAuth 2.0. El token de actualización es un objeto que genera nuevos ID y tokens de acceso cuando los tokens actuales del usuario han caducado. Puede configurar la duración de los tokens de los usuarios en el cliente de aplicación del grupo de usuarios.

También puede elegir la duración de los tokens de actualización. Una vez que caduque el token de actualización de un usuario, este debe volver a iniciar sesión. Si se ha autenticado a través de un IdP SAML, la duración de la sesión de usuario se establece en función de la caducidad de los tokens del usuario y no de la caducidad de la sesión del usuario con su IdP. Su aplicación debe almacenar el token de actualización de cada usuario y renovar su sesión cuando caduque. El inicio de sesión

administrado mantiene las sesiones de los usuarios en una cookie del navegador que es válida durante 1 hora.

Implemente el inicio de IdP-initiated sesión mediante SAML

Al configurar su proveedor de identidad para el inicio de sesión con IdP-initiated SAML 2.0, 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 las afirmaciones de IdP-initiated SAML de un proveedor de identidad externo del grupo de usuarios compatible con el cliente de la aplicación solicitada.



1. Un usuario solicita el inicio de sesión mediante SAML en su aplicación.
2. La aplicación invoca un navegador o redirige al usuario a la página de inicio de sesión del proveedor SAML.
3. El IdP autentica al usuario de forma interactiva o con una sesión recordada en una cookie del navegador.
4. El IdP redirige al usuario a la aplicación con la aserción SAML en el cuerpo de POST.
5. La aplicación añade la aserción SAML al cuerpo de POST de una solicitud, en el punto de conexión `saml2/idpresponse` del grupo de usuarios.

6. Amazon Cognito emite un código de autorización para su usuario.
7. El usuario presenta su código de autorización a la aplicación, que lo intercambia por tokens web JSON (JWT).
8. La aplicación acepta y procesa el token de identificación del usuario como autenticación, genera solicitudes autorizadas a los recursos con un token de acceso y almacena el token de actualización.

Los siguientes pasos describen el proceso general de configuración e inicio de sesión con un proveedor de IdP-initiated SAML 2.0.

1. Cree o designe un grupo de usuarios y un cliente de aplicación.
2. Cree un IdP SAML 2.0 en el grupo de usuarios.
3. Configure su IdP para que admita el inicio del IdP. IdP-initiated SAML introduce consideraciones de seguridad a las que no están sujetos otros proveedores de SSO. Por este motivo, no puedes añadir aplicaciones que no sean de SAML IdPs, ni siquiera el propio grupo de usuarios, a ningún cliente de aplicaciones que utilice un proveedor de SAML al iniciar sesión. IdP-initiated
4. Asocia tu proveedor de IdP-initiated SAML a un cliente de aplicaciones de tu grupo de usuarios.
5. Dirija al usuario a la página de inicio de sesión del IdP SAML y recupere una aserción SAML.
6. Dirija al usuario al punto de conexión `saml2/idpresponse` del grupo de usuarios con la aserción SAML.
7. Reciba los tokens web JSON (JWT).

Para aceptar aserciones de SAML no solicitadas en el grupo de usuarios, debe tener en cuenta su repercusión en la seguridad de la aplicación. Cuando aceptas solicitudes, es probable que se produzcan intentos de suplantación de identidad y CSRF. IdP-initiated Aunque su grupo de usuarios no puede verificar una sesión de inicio de IdP-initiated sesión, Amazon Cognito valida los parámetros de solicitud y las afirmaciones de SAML.

Además, la aserción SAML no debe contener ninguna notificación `InResponseTo` y debe haberse emitido en los seis minutos anteriores.

Debe enviar las solicitudes con SAML a su. IdP-initiated `/saml2/idpresponse` En el caso de las solicitudes de autorización de inicio de sesión gestionadas SP-initiated y gestionadas, 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 `HTTP GET`

las solicitudes. Sin embargo, en el caso de las afirmaciones de IdP-initiated SAML, los detalles de la solicitud deben tener el formato de un `RelayState` parámetro en el cuerpo de la solicitud. `HTTP POST` El cuerpo de la solicitud también debe contener la aserción SAML como parámetro `SAMLResponse`.

El siguiente es un ejemplo de solicitud y respuesta para un IdP-initiated proveedor de SAML.

```
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]
```

Consola de administración de AWS

Para configurar un IdP para SAML IdP-initiated

1. Cree un grupo de usuarios, un cliente de aplicación y un proveedor de identidades de SAML.
2. Desvincule todos los proveedores de identidades de redes sociales y de OIDC del cliente de aplicación, si tiene alguno asociado.
3. Vaya al menú Proveedores sociales y externos de su grupo de usuarios.
4. Edite o añada un proveedor de SAML.
5. En Inicio de sesión con IdP-initiated SAML, selecciona Aceptar SP-initiated y aseveraciones de SAML. IdP-initiated
6. Seleccione Save changes (Guardar cambios).

API/CLI

Para configurar un IdP para SAML IdP-initiated

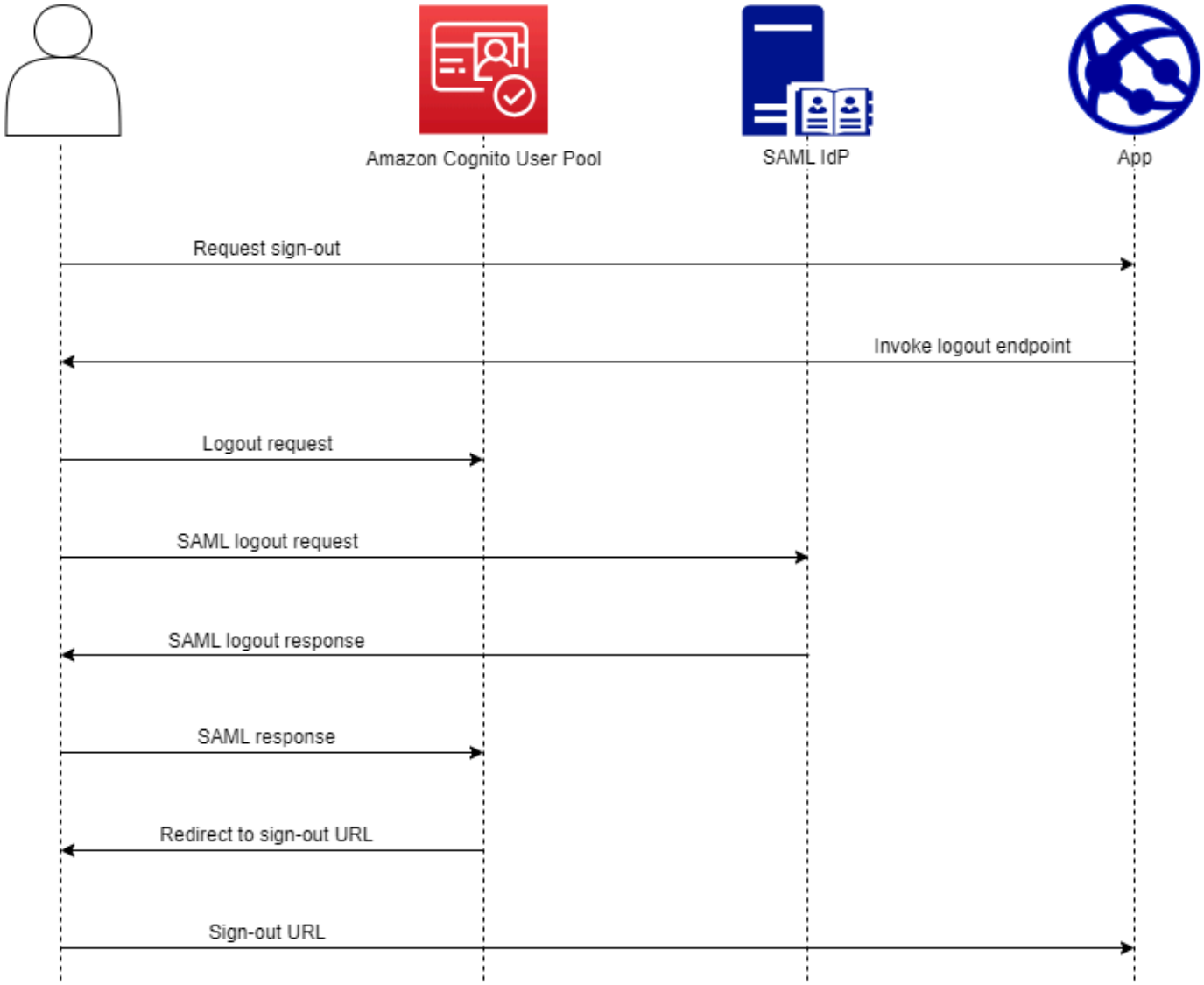
Configure IdP-initiated SAML con el `IDPInit` parámetro de una solicitud de API CreateIdentityProviderо UpdateIdentityProviderуна solicitud. El siguiente es un ejemplo `ProviderDetails` de un IdP que admite IdP-initiated SAML.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Cierre de sesión de usuarios de SAML con un cierre de sesión único

Amazon Cognito admite el cierre de sesión único (SLO) de SAML 2.0. Con el SLO, su aplicación puede cerrar la sesión de los usuarios de sus proveedores de identidad de SAML (IdPs) cuando cierren sesión en su grupo de usuarios. De esta forma, cuando los usuarios quieran volver a iniciar sesión en su aplicación, deberán autenticarse con su IdP SAML. De lo contrario, es posible que tengan cookies del IdP o del navegador de grupo de usuarios que los lleven a su aplicación sin que tengan que proporcionar las credenciales.

Cuando configura su IdP de SAML para que Sign-out admita el flujo, 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 URL de `SingleLogoutService` 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 punto de conexión `/logout` del grupo de usuarios, Amazon Cognito redirige a su usuario de SAML con la siguiente solicitud al punto de conexión de SLO especificado 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 punto de conexión `saml2/logout` con un contenido `LogoutResponse` de su IdP. El IdP debe enviar el contenido `LogoutResponse` en una solicitud `HTTP POST`. A continuación, Amazon Cognito redirige al usuario al destino de redireccionamiento desde la solicitud de cierre de sesión inicial.

Es posible que su proveedor SAML envíe `LogoutResponse` con más de una `AuthnStatement`. El `sessionIndex` en la primera `AuthnStatement` de una respuesta de este tipo debe coincidir con el `sessionIndex` de la respuesta de SAML que ha autentificado originalmente al usuario. Si el `sessionIndex` está en alguna otra `AuthnStatement`, Amazon Cognito no reconocerá la sesión y no se cerrará la sesión del usuario.

Consola de administración de AWS

Configuración de cierre de sesión de SAML

1. Cree un grupo de usuarios, un cliente de aplicación y un IdP SAML.
2. Al crear o editar su proveedor de identidades SAML, en Información del proveedor de identidad, marque la casilla con el título Agregue un flujo de cierre de sesión.
3. En el menú Proveedores sociales y externos de su grupo de usuarios, elija su IdP y busque el Certificado de firma.
4. Seleccione Descargar como .crt.
5. Configure el proveedor 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 redireccionarse a `/saml2/logout` en su dominio del grupo de usuarios.

API/CLI

Para configurar un cierre de sesión de SAML

Configure el cierre de sesión único con el parámetro de una solicitud de API o una solicitud. `IDPSignout` CreateIdentityProviderUpdateIdentityProvider A continuación se muestra un ejemplo de `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

El inicio de sesión con SAML 2.0 se basa en el concepto de que el usuario de la aplicación es el portador de las solicitudes y respuestas de su flujo de autenticación. Le recomendamos que se asegure de que los usuarios no lean ni modifiquen estos documentos de SAML mientras están en tránsito. Para ello, añada la firma y el cifrado de SAML a los proveedores de identidad de SAML (IdPs) de su grupo de usuarios. Con la firma de SAML, los grupos de usuarios añaden una firma a las solicitudes de inicio y cierre de sesión de SAML. Con la clave pública del grupo de usuarios, su IdP puede comprobar que recibe solicitudes de SAML sin modificar. Luego, cuando el IdP responde y pasa las aserciones SAML a las sesiones de navegador de los usuarios, el IdP puede cifrar esa respuesta para que el usuario no pueda inspeccionar sus propios atributos y derechos.

Con la firma y el cifrado de SAML, todas las operaciones criptográficas realizadas durante las operaciones de SAML del grupo de usuarios deben generar firmas y texto cifrado con las claves proporcionadas por el grupo de usuarios que genera Amazon Cognito. Actualmente, no se puede configurar un grupo de usuarios para que firme solicitudes o acepte aserciones cifradas con una clave externa.

> **Note**
>
> Los certificados de su grupo de usuarios tienen una validez de diez 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 se solicita el certificado de firma y firma las solicitudes con el certificado de firma más reciente. Su IdP puede cifrar las aserciones SAML con cualquier certificado de cifrado de grupo de usuarios que no esté caducado. Los certificados anteriores siguen siendo válidos durante toda su duración y la clave pública no cambia de un certificado a otro. Como práctica recomendada, actualice el certificado en la configuración de su proveedor una vez al año.

Temas

- Aceptación de respuestas de SAML cifradas por parte del IdP
- Firma de solicitudes de SAML

Aceptación de respuestas de SAML cifradas por parte del 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 par de claves RSA pública-privada y un certificado a cada proveedor SAML externo que configure en su grupo de usuarios. Al habilitar el cifrado de respuestas para el proveedor SAML de su grupo de usuarios, debe cargar su certificado en un IdP que admita las respuestas de SAML cifradas. Para que la conexión del grupo de usuarios con el IdP SAML funcione, es preciso que previamente el IdP comience a cifrar todas las aserciones SAML con la clave proporcionada.

A continuación, mostramos un flujo de un inicio de sesión de SAML cifrado.

1. El usuario comienza a iniciar sesión y elige su IdP SAML.
2. El Autorizar punto de conexión del grupo de usuarios redirige al usuario a su IdP SAML con una solicitud de inicio de sesión de SAML. Si lo desea, el grupo de usuarios puede acompañar esta solicitud con una firma que permita que el IdP verifique su integridad. Cuando quiera firmar las solicitudes de SAML, debe configurar el IdP para que acepte las solicitudes que su grupo de usuarios haya firmado con la clave pública del certificado de firma.
3. El IdP SAML efectúa el inicio de sesión del usuario y genera una respuesta de SAML. El IdP cifra la respuesta con la clave pública y redirige al usuario al punto de conexión `/saml2/idpresponse` del grupo de usuarios. El IdP debe cifrar la respuesta de acuerdo con la definición incluida en la especificación de SAML 2.0. Para obtener más información, consulte `Element <EncryptedAssertion>` Afirmaciones y protocolos para el lenguaje de marcado de aserciones de seguridad (SAML) de OASIS. V2.0
4. Su grupo de usuarios descifra el texto cifrado de la respuesta de SAML con la clave privada y ejecuta el inicio de sesión del usuario.

> **Important**
>
> Cuando habilita el cifrado de respuestas para un IdP SAML en su grupo de usuarios, el IdP debe cifrar todas las respuestas con una clave pública específica del proveedor. Amazon Cognito no acepta respuestas de SAML que no estén cifradas y provengan de un IdP externo de SAML que configure para posibilitar 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.

Consola de administración de AWS

Configuración del cifrado de respuestas de SAML

1. Cree un grupo de usuarios, un cliente de aplicación y un IdP SAML.
2. Al crear o editar su proveedor de identidades SAML, en Firmar solicitudes y cifrar respuestas, marque la casilla con el título Requerir aserciones SAML cifradas a este proveedor.
3. En el menú Proveedores sociales y externos de su grupo de usuarios, elija su IdP SAML y seleccione Ver certificado de cifrado.
4. Seleccione Descargar como .crt y envíe el archivo descargado a su IdP SAML. Configure su IdP 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 CreateIdentityProvidersolicitud o API. UpdateIdentityProvider A continuación se muestra un ejemplo de `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"
}
```

Para obtener el certificado de cifrado de su grupo de usuarios, realice una solicitud a la DescribeIdentityProviderAPI y recupere el valor del `ActiveEncryptionCertificate` parámetro de respuesta`ProviderDetails`. Guarde este certificado y entréguelo a su IdP como certificado de cifrado para las solicitudes de inicio de sesión de su grupo de usuarios.

Firma de 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 con SAML en Amazon SP-initiated Cognito. Cada grupo de usuarios con un dominio recibe un certificado de firma del grupo de usuarios. X.509 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 SAML. Si lo desea, puede configurar el cliente de aplicación para que firme las solicitudes de inicio de sesión SAML. Al firmar las solicitudes de SAML, el IdP puede comprobar que la firma de los metadatos XML de las solicitudes coincida con la clave pública del certificado del grupo de usuarios que se ha proporcionado.

Consola de administración de AWS

Configuración de la firma de solicitudes de SAML

1. Cree un grupo de usuarios, un cliente de aplicación y un IdP SAML.
2. Al crear o editar el proveedor de identidades de SAML, en Firmar solicitudes y cifrar respuestas, marque la casilla con el título Firmar solicitudes de SAML a este proveedor.
3. En el menú Proveedores sociales y externos de su grupo de usuarios, elija Ver certificado de firma.
4. Seleccione Descargar como .crt y envíe el archivo descargado a su IdP SAML. Configure el IdP SAML para que verifique la firma de las solicitudes de SAML entrantes.

API/CLI

Para configurar la firma de solicitudes de SAML

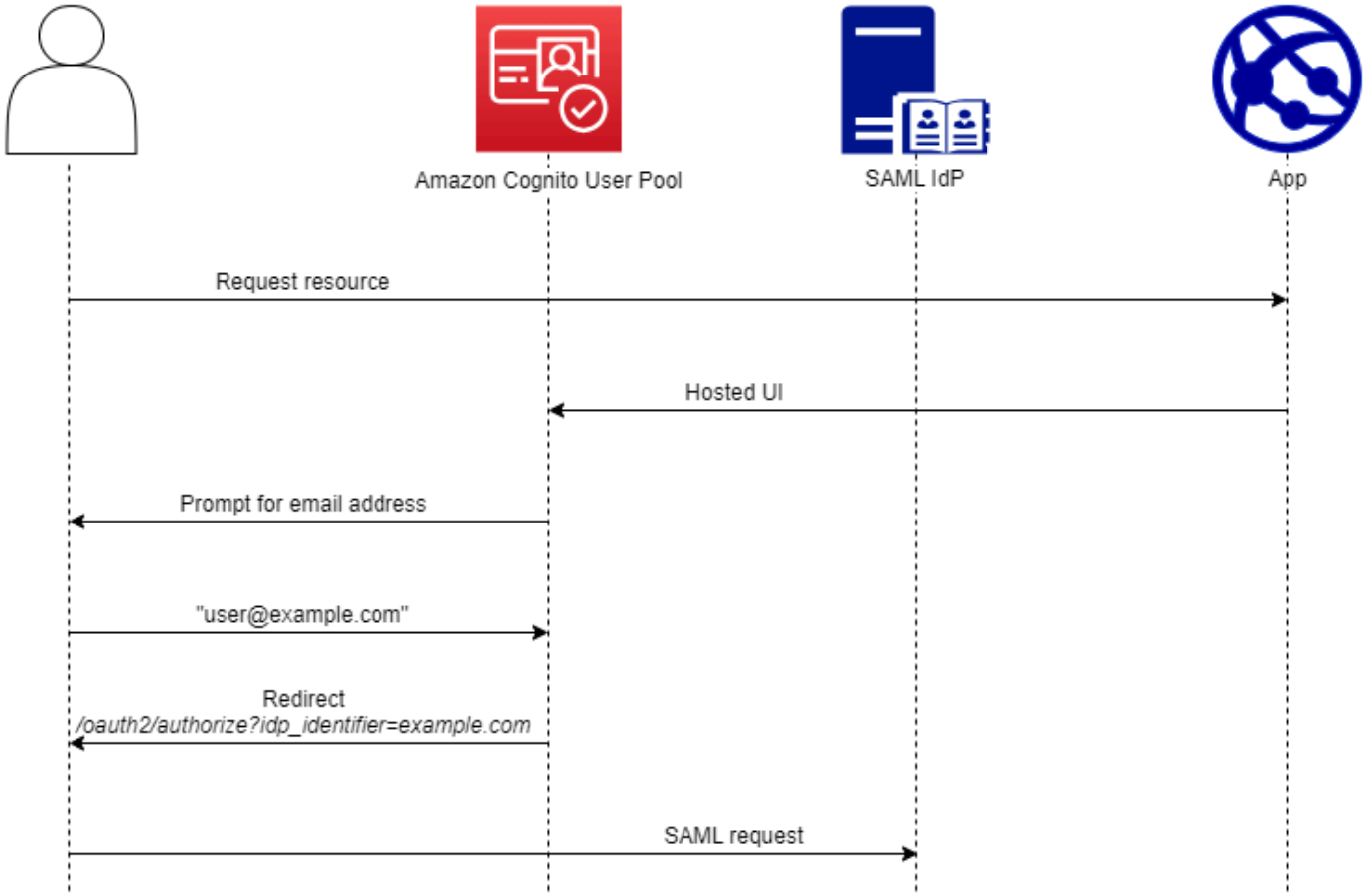
Configure la firma de solicitudes con el `RequestSigningAlgorithm` parámetro de una solicitud CreateIdentityProviderо de UpdateIdentityProviderAPI. A continuación se muestra un ejemplo de `ProviderDetails` de un IdP que admite la firma de solicitudes.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Nombres e identificadores de proveedor de identidades SAML

Al asignar un nombre a los proveedores de identidad de SAML (IdPs) y asignar identificadores de IdP, puede automatizar el flujo SP-initiated de solicitudes de inicio y cierre de sesión dirigidas a ese

proveedor. Para obtener información sobre las restricciones de cadena del nombre del proveedor, consulta la propiedad de. `ProviderName` CreateIdentityProvider



También puede elegir hasta 50 identificadores para los proveedores de SAML. Un identificador es un nombre descriptivo de un IdP de su grupo de usuarios y debe ser único dentro del grupo de usuarios. Si los identificadores de SAML coinciden con los dominios de correo electrónico de los usuarios, el inicio de sesión administrado solicita la dirección de correo electrónico de cada usuario, evalúa el dominio de su dirección de correo electrónico y lo redirige al IdP correspondiente a su dominio. Puesto que la misma organización puede poseer varios dominios, un IdP único puede tener varios identificadores.

Tanto si utiliza identificadores de dominio de correo electrónico como si no, puede utilizar identificadores en una aplicación multiinquilino para redirigir a los usuarios al IdP correcto. Si desea omitir por completo el inicio de sesión administrado, puede personalizar los enlaces que presente a los usuarios para que los redirijan a través del Autorizar punto de conexión a su IdP. Para ejecutar el inicio de sesión de sus usuarios con un identificador y redirigirlos a su IdP, incluya el identificador

con un formato `idp_identifier=myidp.example.com` en los parámetros de la solicitud de autorización inicial.

Otro método para transferir un usuario a su IdP consiste en rellenar el parámetro `identity_provider` con el nombre de su IdP con el siguiente formato de URL.

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
identity_provider=MySAMLIdP&
client_id=1example23456789&
redirect_uri=https://www.example.com
```

Cuando un usuario inicie sesión con su IdP SAML, este lo redirigirá con una respuesta de SAML en el cuerpo de `HTTP POST` a su punto de conexión `/saml2/idpresponse`. Amazon Cognito procesa la aserción SAML y, si las notificaciones de la respuesta cumplen las expectativas, la redirigirá a la URL de devolución de llamada del cliente de aplicación. Cuando el usuario haya completado la autenticación de esta manera, solo interactuará con las páginas web de su IdP y de su aplicación.

Con los identificadores de IdP en formato de dominio, el inicio de sesión administrado 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, supongamos que crea una aplicación que requiere que los empleados de dos empresas diferentes inicien sesión. La primera empresa, A, AnyCompany es propietaria y. `exampleA.com` `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, define el identificador `exampleB.com`.

En su aplicación, invoca el inicio de sesión administrado del cliente de la aplicación para solicitar a cada usuario que introduzca su dirección de correo electrónico. Amazon Cognito obtiene 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 al Autorizar punto de conexión que contiene un parámetro de solicitud `idp_identifier`. Por ejemplo, si un usuario introduce `bob@exampleA.co.uk`, la siguiente página con la que interactuará será 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 la aplicación, puede crear un formulario personalizado que recopile la información introducida por el usuario y la correlacione con el IdP correcto según su propia lógica. Puede generar portales personalizados para cada uno de los inquilinos de la aplicación, de modo que cada uno de ellos enlace 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 el inicio de sesión administrado, asigne al menos un identificador a cada IdP SAML que haya asignado a su cliente de aplicación. De forma predeterminada, la pantalla de inicio de sesión gestionado muestra un botón para cada uno de los botones IdPs que haya asignado a su cliente de aplicación. No obstante, si ha asignado los identificadores correctamente, la página de inicio de sesión de la interfaz de usuario alojada clásica se parecerá a la siguiente imagen.
Una página de inicio de sesión en el inicio de sesión administrado de Amazon Cognito, con el inicio de sesión del usuario local y un aviso para que un usuario federado ingrese una dirección de correo electrónico.

> **Note**
>
> En la interfaz de usuario alojada clásica, la página de inicio de sesión del cliente de la aplicación solicita automáticamente una dirección de correo electrónico cuando le asignas identificadores. IdPs En la experiencia de inicio de sesión administrado, debe habilitar este comportamiento en el editor de marca. En la categoría de configuración de Comportamiento de autenticación, seleccione Entrada de búsqueda de dominio bajo el encabezado Pantalla de proveedor.

Para analizar dominios en el inicio de sesión administrado, deberá utilizar los dominios como identificadores de IdP. Si asignas un identificador de cualquier tipo a cada uno de los SAML IdPs de un cliente de aplicación, el inicio de sesión gestionado de esa aplicación ya no mostrará botones. IdP-selection 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 tus páginas de inicio de sesión gestionadas muestren una lista de ellos IdPs, no asignes identificadores y utilices el parámetro de `identity_provider` solicitud en tus solicitudes de autorización.

- Si asigna solo un IdP SAML a su cliente de aplicación, la página de inicio de sesión del inicio de sesión administrado mostrará un botón para iniciar sesión con ese IdP.

- Si asigna un identificador a cada IdP SAML que active para el cliente de aplicación, aparecerá una petición para que el usuario introduzca una dirección de correo electrónico en la página de inicio de sesión del inicio de sesión administrado.
- Si tiene varios IdPs y no les asigna un identificador a todos, la página de inicio de sesión gestionado muestra un botón para iniciar sesión con cada IdP asignado.
- Si ha asignado identificadores a sus páginas de inicio de sesión gestionadas IdPs y desea que muestren 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 ya existente y agregarlo de nuevo sin identificador. Si crea un nuevo IdP, los usuarios de SAML crean nuevos perfiles de usuario. Esta duplicación de los usuarios activos puede repercutir en la facturación del mes en el que cambie la configuración del IdP.

Para obtener más información sobre la configuración de IdP, consulte Configuración de proveedores de identidad para su grupo de usuarios.

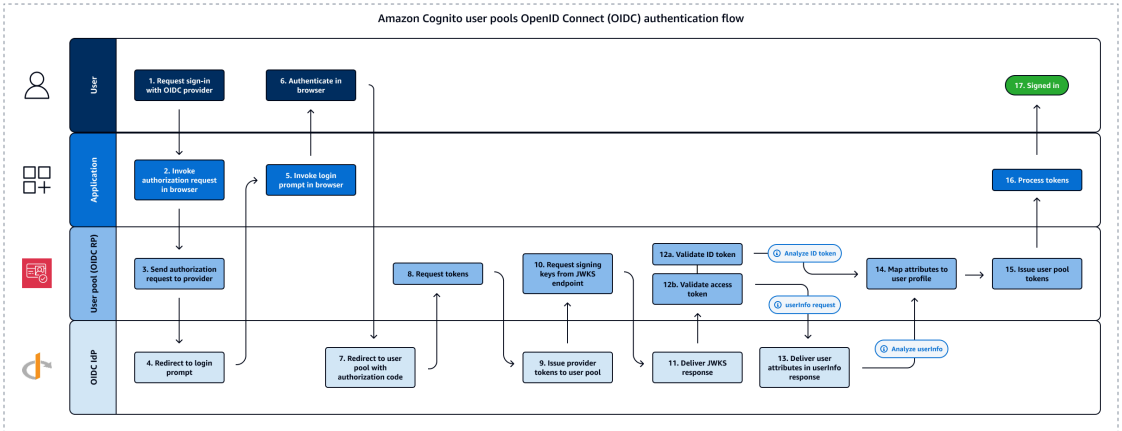
Uso de proveedores de identidades de OIDC con un grupo de usuarios

Los usuarios pueden iniciar sesión en la aplicación con sus cuentas existentes de los proveedores de identidad de OpenID Connect (OIDC) (). IdPs Con los proveedores OIDC, los usuarios de sistemas de inicio de sesión único independientes pueden proporcionar credenciales que ya existen, mientras que su aplicación recibe los tokens de OIDC en el formato compartido de grupos de usuarios. Para configurar un IdP de OIDC, configure su IdP para que gestione su grupo de usuarios como RP y configure su aplicación para que gestione su grupo de usuarios como IdP. Amazon Cognito sirve como paso intermedio entre varios OIDC IdPs y sus aplicaciones. Su grupo de usuarios aplica reglas de asignación de atributos a las reclamaciones que figuran en el identificador y los tokens de acceso que su proveedor transfiere directamente a su grupo de usuarios. A continuación, Amazon Cognito emite nuevos tokens en función de los atributos de usuario asignados y de cualquier ajuste adicional que haya realizado en el flujo de autenticación con los desencadenadores de Lambda.

Los usuarios que inician sesión con un IdP OIDC no están obligados a proporcionar credenciales ni información nuevas para acceder a la aplicación de su grupo de usuarios. La aplicación puede redirigirlos silenciosamente a su IdP para iniciar sesión, con un grupo de usuarios como herramienta en segundo plano que estandariza el formato de token de la aplicación. Para obtener más información sobre la redirección del IdP, consulte Autorizar punto de conexión.

Al igual que ocurre con otros proveedores de identidad de terceros, debe registrar su aplicación en el proveedor de OIDC y obtener información sobre la aplicación de IdP que desea conectar a su

grupo de usuarios. Un IdP OIDC de un grupo de usuarios requiere un ID de cliente, un secreto de cliente, los ámbitos que desee solicitar e información sobre los puntos de conexión del servicio del proveedor. Su grupo de usuarios puede descubrir los puntos de conexión de OIDC del proveedor desde un punto de conexión de detección o puede introducirlos manualmente. También debe examinar los tokens de ID de proveedores y crear asignaciones de atributos entre el IdP y los atributos de su grupo de usuarios.



Consulte Flujo de autenticación de proveedores de identidad (IdP) de grupos de usuarios OIDC para obtener más información sobre este flujo de autenticación.

> **Note**
>
> Sign-in a través de un tercero (federación) está disponible en los grupos de usuarios de Amazon Cognito. Esta característica es independiente de la federación de OIDC con el grupos de identidades de Amazon Cognito.

Puede añadir un IdP de OIDC a su grupo de usuarios mediante el método API del Consola de administración de AWS grupo de usuarios AWS CLI, o mediante él. CreateIdentityProvider

Temas

- Requisitos previos
- Registro de una solicitud con un IdP OIDC
- Agregar un IdP OIDC al grupo de usuarios
- Probar la configuración del proveedor de identidades (IdP) OIDC
- Flujo de autenticación de proveedores de identidad (IdP) de grupos de usuarios OIDC

- Step-up autenticación para usuarios federados

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 los 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.
  - Presenta una clave pública no caducada con una cadena de confianza de CA de raíz válida.

Registro de una solicitud con un IdP OIDC

Antes de añadir un IdP OIDC a la configuración de su grupo de usuarios y asignarlo a los clientes de aplicación, debe configurar una aplicación de cliente OIDC en su IdP. Su grupo de usuarios es la aplicación de la parte que depende y que administra la autenticación con su IdP.

Para registrarse en un proveedor de identidad OIDC

1. Crear una cuenta de desarrollador con el proveedor de identidad OIDC.

Enlaces a OIDC IdPs

Proveedor de identidad OIDC	¿Cómo instalarlo	URL de descubrimiento del OIDC
Salesforce	Salesforce as an OpenID Connect Identity Provider	`https://MyDomainName.my.salesforce.com/.well-known/openid-configuration`
OneLogin	Connect an OIDC enabled app	`https://your-domain.onelogin.com/oidc/2/.well-known/openid-configuration`
JumpCloud	SSO with OIDC	`https://oauth.id.jumpcloud.com/.well-known/openid-configuration`
Okta	Instalar un proveedor de identidad Okta	`https://Your Okta subdomain.okta.com/.well-known/openid-configuration`
ID de Microsoft Entra	OpenID Connect on the Microsoft identity platform	`https://login.microsoftonline.com/{tenant}/v2.0` Los valores de `tenant` pueden incluir un ID de inquilino, `common`, `organizations` o `consumers`.

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.auth.us-east-1.amazoncognito.com/oauth2/idpresponse
```

3. Seleccione los ámbitos que desee que su directorio de usuarios compartan con su grupo de usuarios. El scope openid es obligatorio para que OIDC IdPs pueda ofrecer cualquier información de usuario. El ámbito `email` concede acceso a las reclamaciones `email` y `email_verified`. Los ámbitos adicionales de la especificación OIDC son `profile` para todos los atributos de usuario y phone para `phone_number` y `phone_number_verified`.
4. El proveedor de identidad OIDC le proporciona un ID y un secreto de cliente. Anote estos valores y agréguelos a la configuración del IdP OIDC que agregue a su grupo de usuarios más adelante.

Ejemplo: Utilizar Salesforce como un proveedor de identidad OIDC con el grupo de usuarios

Utiliza un IdP de OIDC cuando quiere establecer confianza entre un OIDC-compatible IdP como Salesforce y su 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 ve Configuración en el encabezado, elija su nombre en la barra de navegación superior y elija 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.auth.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.auth.us-east-1.amazoncognito.com/oauth2/idpresponse
```

9. Seleccione los ámbitos. Debe incluir el valor openid de ámbito. Para conceder acceso a las reclamaciones email y 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.

Agregar un IdP OIDC al grupo de usuarios

Después de configurar el IdP, puede configurar el grupo de usuarios para que gestione las solicitudes de autenticación con un IdP OIDC.

Amazon Cognito console

Adición de un IdP OIDC en la consola

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. Seleccione el menú Proveedores sociales y externos y, a continuación, seleccione Agregar un proveedor de identidades.
5. Elija un IdP OpenID Connect.
6. Introduzca un Nombre de proveedor único.
7. Escriba el ID de cliente del IdP. Este es el ID del cliente de aplicación que ha creado en su IdP OIDC. El ID de cliente que proporcione debe ser un proveedor OIDC que haya configurado con una URL de devolución de llamada de `https://[your user pool domain]/oauth2/idpresponse`.
8. Introduzca el Secreto de cliente del IdP. Debe ser el secreto de cliente del mismo cliente de aplicación del paso anterior.
9. Introduzca los Ámbitos autorizados para este proveedor. Los ámbitos definen los grupos de atributos de usuario (por ejemplo, `name` y`email`) que la aplicación solicita a su proveedor. Los ámbitos deben estar separados por espacios, de acuerdo con la especificación OAuth 2.0.

   Es posible que el IdP solicite a los usuarios que estén de acuerdo con proporcionar estos atributos a su aplicación al iniciar sesión.
10. Elija un método de solicitud de atributos. IdPspuede requerir que las solicitudes a sus `userInfo` puntos finales tengan el formato de o`GET`. `POST` El punto de conexión `userInfo` de Amazon Cognito requiere solicitudes `HTTP GET`, por ejemplo.
11. Elija un método de configuración según la forma en que desee que su grupo de usuarios determine la ruta a los OIDC-federation puntos finales clave de su IdP. Por lo general, IdPs aloje un `/well-known/openid-configuration` punto final en la URL base del emisor. Si este es el caso de su proveedor, la opción Completar automáticamente a través de la URL del emisor le solicitará la URL base, intentará acceder a la ruta `/well-known/openid-configuration` desde allí y leerá los puntos de conexión que aparezcan. Es posible que tenga rutas de punto de conexión atípicas o que desee pasar las solicitudes a uno o más puntos de conexión a través de un proxy alternativo. En ese caso, seleccione Entrada manual y especifique las rutas para los puntos de conexión `authorization`, `token`, `userInfo` y `jwks_uri`.

   > **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.

12. Configure sus reglas de asignación de atributos en Asignar atributos entre su proveedor de OpenID Connect y su grupo de usuarios. El atributo del grupo de usuarios es el atributo de destino del perfil de usuario de Amazon Cognito y el atributo OpenID Connect es el atributo de origen que desea que Amazon Cognito busque en una reclamación o respuesta. ID-token `userInfo` Amazon Cognito asigna automáticamente la reclamación OIDC del tipo sub en `username` en el perfil de usuario de destino.

    Para obtener más información, consulte Asignación de atributos de IdP a perfiles y tokens.
13. Elija Agregar proveedores de identidades.
14. En el menú Clientes de aplicación, seleccione un cliente de aplicación de la lista. Vaya a la pestaña Páginas de inicio de sesión y, en Configuración de páginas de inicio de sesión administrado, seleccione Editar. Busque los proveedores de identidad y añada su nuevo IdP OIDC.
15. Seleccione Save changes (Guardar cambios).

API/CLI

Consulte la configuración del OIDC en el segundo ejemplo en. CreateIdentityProvider Puede modificar esta sintaxis y utilizarla como cuerpo de la solicitud de `CreateIdentityProvider` o `UpdateIdentityProvider`, o como el archivo de entrada `--cli-input-json` para create-identity-provider.

Probar la configuración del proveedor de identidades (IdP) OIDC

En su aplicación, debe invocar un navegador en el cliente del usuario para que pueda iniciar sesión con su proveedor OIDC. Pruebe el inicio de sesión con su proveedor después de haber completado los procedimientos de configuración de las secciones anteriores. El siguiente ejemplo de URL carga la página de inicio de sesión de su grupo de usuarios con un dominio de prefijo.

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Este enlace es la página a la que Amazon Cognito lo dirige cuando va al menú Clientes de aplicación, selecciona un cliente de aplicación, navega a la pestaña Páginas de inicio de sesión y selecciona Ver página de inicio de sesión. Para obtener más información sobre los dominios del grupo de usuarios, consulte Configuración de un dominio del grupo de usuarios. Para obtener más información acerca de los clientes de aplicación, incluidos los ID de cliente y las URL de respuesta, consulte Application-specific configuración con clientes de aplicaciones.

El siguiente enlace de ejemplo configura la redirección silenciosa al proveedor `MyOIDCIdP` desde el Autorizar punto de conexión con el parámetro de consulta `identity_provider`. Esta URL omite el inicio de sesión interactivo del grupo de usuarios con el inicio de sesión administrado y va directamente a la página de inicio de sesión del IdP.

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
identity_provider=MyOIDCIdP&response_type=code&client_id=1example23456789&redirect_uri=https://
www.example.com
```

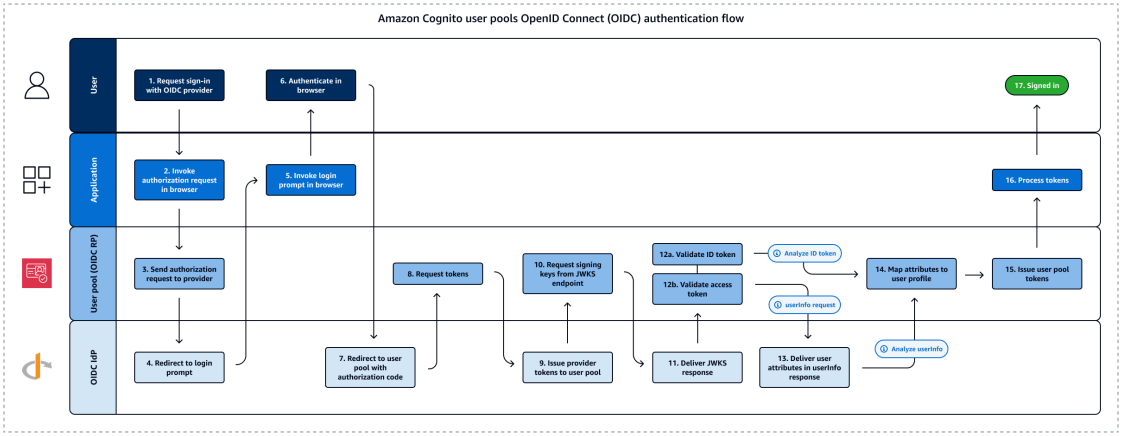
Flujo de autenticación de proveedores de identidad (IdP) de grupos de usuarios OIDC

Con el inicio de sesión con OpenID Connect (OIDC), un grupo de usuarios automatiza el flujo de inicio de sesión con código de autorización con su proveedor de identidades (IdP). Una vez que el usuario haya completado el inicio de sesión con su IdP, Amazon Cognito toma su código en el punto de conexión `oauth2/idpresponse` del proveedor externo. Con el token de acceso obtenido, el grupo de usuarios consulta el punto de conexión `userInfo` del IdP para recuperar los atributos del usuario. A continuación, el grupo de usuarios compara los atributos recibidos con las reglas de asignación de atributos que se han configurado y rellena el perfil del usuario y el token de identificación en consecuencia.

Los ámbitos de OAuth 2.0 que solicita en la configuración del proveedor de OIDC definen los atributos de usuario que el IdP proporciona a Amazon Cognito. Como práctica de seguridad recomendada, solicite únicamente los ámbitos que correspondan a los atributos que desee asignar al grupo de usuarios. Por ejemplo, si el grupo de usuarios solicita `openid profile`, obtendrá todos los atributos posibles, pero si se solicita `openid email phone_number`, solo obtendrá la dirección de correo electrónico y el número de teléfono del usuario. Puede configurar los ámbitos que solicita al OIDC IdPs para que difieran de los que autoriza y solicita en la solicitud de autenticación del cliente de la aplicación y del grupo de usuarios.

Cuando un usuario inicia sesión en su aplicación a través de un IdP OIDC, su grupo de usuarios lleva a cabo el siguiente flujo de autenticación.

1. Un usuario accede a su página de inicio de sesión administrado y elige iniciar sesión con su IdP OIDC.
2. La aplicación dirige el navegador del usuario al punto de conexión de autorización de su grupo de usuarios.
3. El grupo de usuarios redirige la solicitud al punto de conexión de autorización del IdP OIDC.
4. Su IdP muestra una petición de inicio de sesión.
5. En su aplicación, la sesión de usuario muestra una petición de inicio de sesión para el IdP OIDC.
6. El usuario introduce sus credenciales para el IdP o presenta una cookie para una sesión ya autenticada.
7. Una vez que el usuario se haya autentica, el IdP OIDC lo redirige a Amazon Cognito con un código de autorización.
8. Su grupo de usuarios intercambia el código de autorización por tokens de identificación y acceso. Amazon Cognito recibe los tokens de acceso cuando configura su IdP con los ámbitos `openid`. Las aserciones del token de ID y la respuesta a `userInfo` vienen determinadas por ámbitos adicionales de la configuración de su IdP, como `profile` y `email`.
9. Su IdP emite los tokens solicitados.
10. Su grupo de usuarios determina la ruta al punto de conexión de `jwks_uri` del IdP a partir de las URL del emisor en su configuración de IdP y solicita las claves de firma del token desde el punto de conexión del conjunto de claves web JSON (JWKS).
11. El IdP devuelve las claves de firma desde el punto de conexión de JWKS.
12. Su grupo de usuarios valida los tokens de IdP a partir de los datos de firma y caducidad de los tokens.
13. Su grupo de usuarios autoriza una solicitud al punto de conexión`userInfo` del IdP con el token de acceso. El IdP responde con los datos del usuario en función de los ámbitos del token de acceso.
14. Su grupo de usuarios compara el token de ID y la respuesta a `userInfo` del IdP con las reglas de asignación de atributos de su grupo de usuarios. Escribe los atributos de IdP asignados en los atributos del perfil del grupo de usuarios.
15. Además, otorga a la aplicación tokens de portador, que pueden incluir tokens de identidad, acceso y actualización:
16. La aplicación procesa los tokens del grupo de usuarios e inicia la sesión del usuario.



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

El OIDC es una capa de identidad adicional a la de OAuth 2.0, que especifica los tokens de identidad JSON-formatted (JWT) que emiten las aplicaciones cliente del OIDC (partes que dependen). 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, el inicio de sesión administrado 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. Amazon Cognito actualiza la clave de firma desde el punto de conexión JWKS en su configuración de IdP para cada token de ID de IdP que procese.
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.

Step-up autenticación para usuarios federados

Para los usuarios federados que inician sesión a través de un proveedor de identidad (IdP) de OpenID Connect (OIDC), Amazon Cognito utiliza los valores ACR y AMR del IdP de OIDC en lugar de calcular los suyos propios. El IdP del OIDC es propietario de la autenticación de estos usuarios federados. Step-up la autenticación de un usuario federado se realiza a través del IdP del OIDC: Amazon Cognito redirige al IdP con. `acr_values` Para obtener más información sobre los

niveles de ACR y los valores de AMR, consulte. the section called “Niveles de autenticación con notificaciones de ACR y AMR”

Los valores de ACR del IdP del OIDC se asignan a los niveles de Amazon Cognito con `AcrMapping` el campo del IdP. Configure este campo con la operación o. CreateIdentityProviderUpdateIdentityProvider Amazon Cognito asigna el valor ACR del IDP de OIDC a un nivel de Amazon Cognito y, a continuación, estampa el nombre del nivel correspondiente que configuró para su grupo de usuarios en el token de Amazon Cognito. El siguiente ejemplo configura un para un proveedor de OIDC. `AcrMapping`

```
{
  "ProviderName": "MyOktaIdP",
  "ProviderType": "OIDC",
  "AcrMapping": {
    "Level1": "urn:okta:loa:1",
    "Level2": "urn:okta:loa:2",
    "Level3": "urn:okta:loa:3",
    "Level4": "urn:okta:loa:4"
  }
}
```

La configuración `AcrMapping` del proveedor de identidad está disponible en todos los planes de funciones. No se limita al plan de funciones Essentials o Plus.

El siguiente comportamiento se aplica a los usuarios federados que inician sesión a través de un IdP de OIDC.

- Si el IDP del OIDC devuelve un valor de ACR que no está en el configurado, Amazon `AcrMapping` Cognito establece la `acr` afirmación del token en el nivel más bajo (el predeterminado o el nombre de nivel 1 que `urn:cognito:loa:1` configuró para su grupo de usuarios). No se trata de un error.
- Amazon Cognito elimina cualquier valor de AMR que no reconozca y conserva solo los valores conocidos.
- Si el IdP del OIDC no admite ACR y AMR, o no devuelve ningún valor de AMR que Amazon Cognito reconozca, Amazon Cognito lo establece como valor AMR de marcador de posición predeterminado. `fed`
- Amazon Cognito confía en los `amr` valores `acr` y valores que afirma el IdP del OIDC, sujetos a mapeo y filtrado. No valida qué desafíos realizó realmente el IdP.

En un flujo de federación, la aplicación se redirige al `/oauth2/authorize` punto final con el nivel o los niveles objetivo expresados en los nombres de su grupo de usuarios. Amazon Cognito traduce los valores en los valores que el IdP del OIDC espera recibir, los redirige al IdP`AcrMapping`, recibe los valores ACR y AMR del IdP, los mapea y emite los tokens de Amazon Cognito. Para obtener más información sobre la configuración de un IdP, consulte Configuración de proveedores de identidad.

Asignación de atributos de IdP a perfiles y tokens

Los servicios de proveedores de identidades (IdP), incluido Amazon Cognito, suelen registrar más información sobre un usuario. Puede que quiera saber para qué empresa trabaja, cómo ponerse en contacto con él y otra información de identificación. Sin embargo, el formato que adoptan estos atributos varía según los proveedores. Por ejemplo, configure tres IdPs de tres proveedores diferentes con su grupo de usuarios y examine un ejemplo de aserción, token de ID o `userInfo` carga útil de SAML de cada uno. Uno representará la dirección de correo electrónico del usuario como `email`, otro como `emailaddress` y el tercero como `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`.

Una de las principales ventajas de la consolidación IdPs con un grupo de usuarios es la posibilidad de mapear la variedad de nombres de atributos en un único esquema de token OIDC con nombres de atributos compartidos, predecibles y consistentes. De esta forma, los desarrolladores no tendrán que mantener la lógica para procesar varios eventos de inicio de sesión único complejos. Esta consolidación de formatos es la asignación de atributos. La asignación de atributos del grupo de usuarios asigna los nombres de los atributos del IdP a los nombres de los atributos del grupo de usuarios correspondientes. Por ejemplo, puede configurar el grupo de usuarios para que escriba el valor de una notificación `emailaddress` para el atributo estándar `email` del grupo de usuarios.

Cada IdP de grupo de usuarios tiene un esquema de asignación de atributos independiente. Para especificar las asignaciones de atributos para su IdP, configure un proveedor de identidades de grupos de usuarios en la consola de Amazon Cognito, un AWS SDK o la API de REST de grupos de usuarios.

Cosas que debe saber acerca de los asignaciones

Antes de empezar a configurar la asignación de atributos del usuario, revise la siguiente información importante.

- Para poder asignar una notificación de IdP a un atributo personalizado, ese atributo personalizado debe existir ya en el esquema de su grupo de usuarios. Los atributos personalizados se definen al crear el grupo de usuarios o se añaden posteriormente, y no se puede asignar un atributo de

IdP a un atributo personalizado que aún no se haya agregado al esquema. Para obtener más información sobre cómo definir los atributos del grupo, consulteUso de atributos de usuario.

- 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 obtiene el atributo `username` en el perfil de un usuario federado a partir de notificaciones específicas que el IdP federado aprueba, como se muestra en la siguiente tabla. Amazon Cognito anexa este valor de atributo al nombre de su IdP, por ejemplo `MyOIDCIdP_[sub]`. Cuando desee que los usuarios federados tengan un atributo que coincida exactamente con un atributo del directorio externo, asigne dicho atributo a un atributo de inicio de sesión de Amazon Cognito como `preferred_username`.

Proveedor de identidades	Atributo de origen de **`username`**
Facebook	`id`
Google	`sub`
Login with Amazon	`user_id`

Proveedor de identidades	Atributo de origen de **`username`**
Inicio de sesión con Apple	`sub`
Proveedores SAML	`NameID`
Proveedores de OpenID Connect (OIDC)	`sub`

- Cuando un grupo de usuarios no distingue entre mayúsculas y minúsculas, Amazon Cognito convierte todo el nombre de usuario federado generado automáticamente a minúsculas, incluido el prefijo del nombre del IdP. A continuación se muestra un ejemplo de nombre de usuario para un grupo de usuarios que distingue entre mayúsculas y minúsculas: `MySAML_TestUser@example.com`. El siguiente es el mismo nombre de usuario para un grupo de usuarios que no distingue entre mayúsculas y minúsculas: `mysaml_testuser@example.com`.

  En los grupos de usuarios que no distinguen mayúsculas de minúsculas, los activadores de Lambda que procesan el nombre de usuario deben tener en cuenta los cambios en todo el nombre de usuario. Estos cambios incluyen cualquier valor en mayúsculas y minúsculas de los atributos de origen del nombre de usuario que haya mapeado desde su IdP. Para vincular su IdP a un grupo de usuarios con una configuración de distinción entre mayúsculas y minúsculas diferente a la del grupo actual, cree un grupo de usuarios nuevo.
- 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 solo actualiza los atributos mapeados cuando sus valores cambian. 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 añada al seleccionar Añadir atributos personalizados en el menú. Sign-up 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 el 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 el IdP envía un valor para un atributo asignado 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 encerrada entre los caracteres entre corchetes `[` y `]`. El formulario URL de Amazon Cognito codifica los valores que contienen caracteres no alfanuméricos excepto `.`, `-`, * y `_`. Debe descodificar y analizar los valores individuales antes de usarlos en la aplicación.
- El atributo de destino conserva cualquier valor que le asignen las reglas de asignación de atributos, a menos que lo modifique una acción administrativa o de inicio de sesión. Amazon Cognito no elimina los atributos de los usuarios cuando el atributo de origen ya no se envía en el token del proveedor o en la aserción de SAML. Las siguientes acciones eliminan el valor de un atributo del perfil de un grupo de usuarios para un usuario federado:
  1. El IdP envía un valor en blanco para el atributo de origen y una regla de asignación aplica el valor en blanco al atributo de destino.
  2. Borra el valor del atributo mapeado con una AdminDeleteUserAttributessolicitud DeleteUserAttributeso.

Especificar las asignaciones de atributos de los proveedores de identidad para su grupo de usuarios (Consola de administración de AWS)

Puede usarlo Consola de administración de AWS para especificar las asignaciones de atributos para el IdP de su grupo de usuarios.

> **Note**
>
> Amazon Cognito mapea las notificaciones entrantes con los atributos del grupo de usuarios solo si las notificaciones existen en el token entrante. Si una notificación asignada anteriormente ya no existe en el token de entrada, no cambiará ni se eliminará. Si tu aplicación requiere el mapeo de las notificaciones eliminadas, puedes usar el activador Pre-Authentication Lambda para eliminar el atributo personalizado durante la autenticación y permitir que estos atributos se rellenen desde el token entrante.

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. Seleccione el menú Proveedores sociales y externos.
4. Elija 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 Uso de proveedores de identidades de redes sociales 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 de ***<provider>***atributos. Este es el atributo transferido 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. Seleccione Save changes (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. Seleccione el menú Proveedores sociales y externos.
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 identidades 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 es el atributo transferido 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 URL-formatted atributos similares a:

      ```
      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. Seleccione Save changes (Guardar cambios).

Especificar las asignaciones de atributos de los proveedores de identidad para su grupo de usuarios (AWS CLI and AWS API)

El siguiente cuerpo de la solicitud CreateIdentityProvidercorresponde o UpdateIdentityProviderasigna los atributos `emailaddress` «MyIdP» del proveedor de SAML y `phone` a los atributos del grupo de usuarios `email`, y`birthdate`, `phone_number`, en ese orden. `birthdate` Este es el cuerpo completo de la solicitud para un proveedor de SAML 2.0; el cuerpo de la solicitud varía según el tipo de IdP y los detalles específicos. La asignación de atributos está en el parámetro `AttributeMapping`.

```
{
   "AttributeMapping": {
      "email" : "emailaddress",
      "birthdate" : "birthdate",
      "phone_number" : "phone"
   },
   "IdpIdentifiers": [
      "IdP1",
      "pdxsaml"
   ],
   "ProviderDetails": {
```

```
      "IDPInit": "true",
      "IDPSignout": "true",
      "EncryptedResponses" : "true",
      "MetadataURL": "https://auth.example.com/sso/saml/metadata",
      "RequestSigningAlgorithm": "rsa-sha256"
   },
   "ProviderName": "MyIdP",
   "ProviderType": "SAML",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

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 *<SAML metadata XML>* contiene comillas (), deben estar " ocultas (). \"

  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
```

- API/SDK: 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`
- API/SDK: 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>`
- API/SDK: 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 uniforme en tu 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 este realice su primer inicio de sesión, en una solicitud de API `AdminLinkProviderForUser`, bien en una tarea de fase previa planificada

o en un Desencadenador de Lambda Antes del registro.. 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 este modo, puede configurar el perfil local con las solicitudes de acceso y mantener actualizadas sus solicitudes de identidad con su proveedor. 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 haya creado y, a continuación, haya vinculado a una identidad de usuario de terceros tiene el nombre de usuario con el que se ha creado y un atributo `identities` 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 IdPs ADFS1, ADFS2 y ADFS3.

> **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 los usuarios a cada IdP desde un máximo de cinco notificaciones de atributos de IdP, tal y como se define en el parámetro `ProviderAttributeName` de `SourceUser` en una solicitud de API `AdminLinkProviderForUser`. Por ejemplo, si ha vinculado al menos un usuario a los atributos de origen `email`, `phone`, `department`, `given_name` y `location`, solo puede vincular usuarios adicionales en uno de esos cinco atributos.
- Puede vincular usuarios federados a un perfil de usuario federado existente o a un usuario local.
- No puede vincular a los proveedores a los perfiles de usuario del. Consola de administración de AWS
- El token de ID de usuario contiene todos sus proveedores asociados en el`identities`notificación.
- Puede establecer una contraseña para el perfil de usuario federado creado automáticamente en una solicitud de AdminSetUserPasswordAPI. 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 por 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 notificaciones de identidad

en el token de ID de un IdP OIDC y también comprueba el punto de conexión `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.

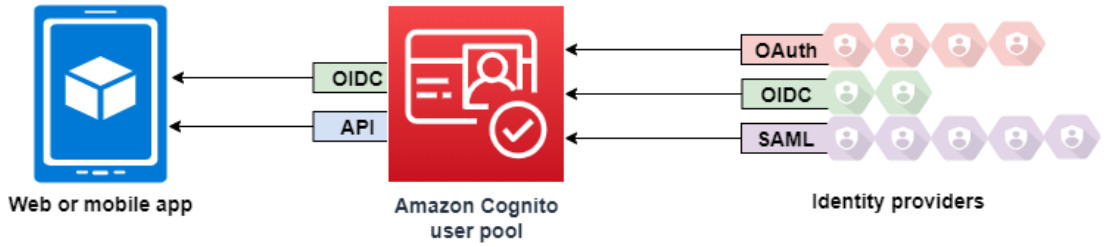
Acceso a los recursos después de iniciar sesión correctamente

Los usuarios de la aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidades (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 Descripción de los tokens web JSON (JWT) de los grupos de usuarios.

Tras una autenticación correcta, la aplicación recibe los tokens del grupo de usuarios de Amazon Cognito. Puede usar los tokens de 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.
- Proporcionar una prueba de autenticación temporal revocable.
- Introducir los datos de identidad en un perfil de usuario de la aplicación.
- Autorizar los cambios en el perfil del usuario conectado en el directorio del grupo de usuarios.
- Autorizar las solicitudes de información de usuario con un token de acceso.
- Autorizar las solicitudes a los datos que se encuentran detrás de las API externas protegidas contra acceso mediante tokens de acceso.
- Autorizar el acceso a los recursos de la aplicación almacenados en el cliente o el servidor con Amazon Verified Permissions.

Para obtener más información, consulte Un ejemplo de sesión de autenticación y Descripción de los tokens web JSON (JWT) de los grupos de usuarios.



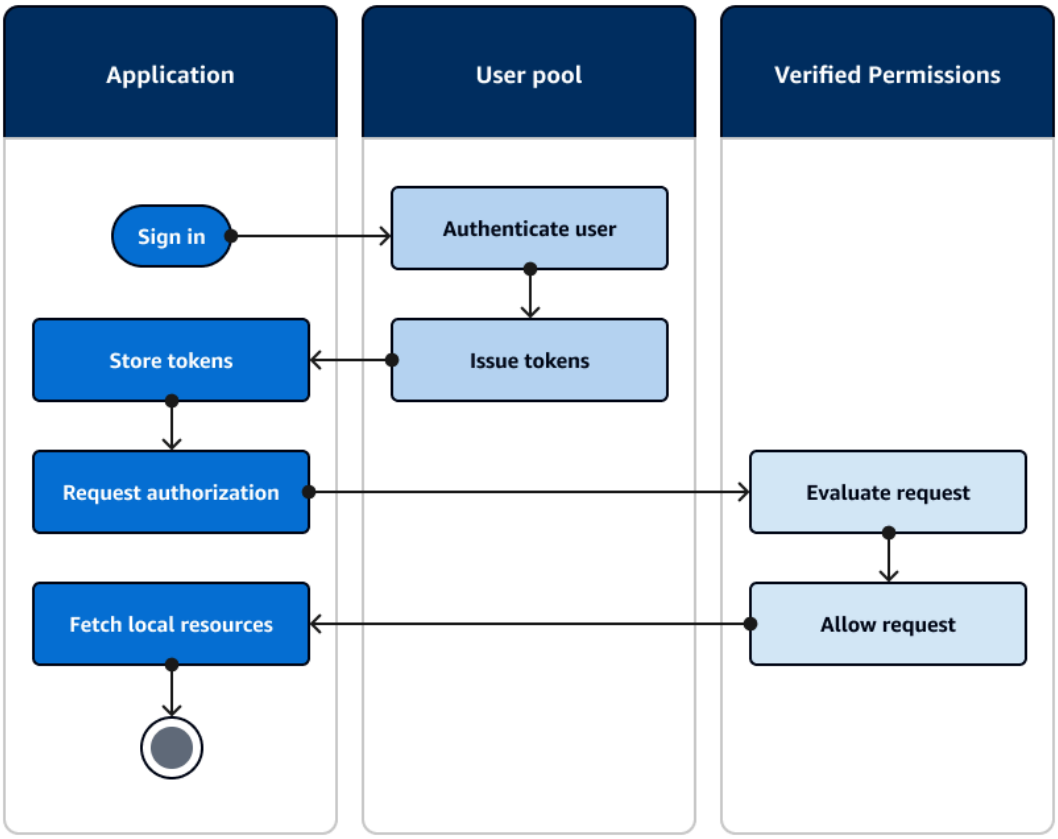
Temas

- Autorización del acceso a los recursos del cliente o del servidor con Amazon Verified Permissions
- Acceso a recursos con API Gateway después del inicio de sesión
- Acceso Servicios de AWS usar un grupo de identidades después de iniciar sesión

Autorización del acceso a los recursos del cliente o del servidor con Amazon Verified Permissions

La aplicación puede transferir a Amazon Verified Permissions los tokens de un usuario que haya iniciado sesión. Permisos verificados es un servicio de autorización y administración de permisos escalable y detallado para las aplicaciones que ha creado. Un grupo de usuarios de Amazon Cognito puede ser un origen de identidad para un almacén de políticas de Verified Permissions. Verified Permissions toma las decisiones de autorización respecto a las acciones y los recursos solicitados, como `GetPhoto` para `premium_badge.png`, basándose en la entidad 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.



Introducción a Amazon Verified Permissions

Una vez que haya integrado el grupo de usuarios con Verified Permissions, tendrá un origen centralizado para autorizaciones detalladas para todas las aplicaciones de Amazon Cognito. Esto elimina la necesidad de disponer de una lógica de seguridad de gran precisión que, si debiera aplicarse, sería preciso codificar y replicar entre todas sus aplicaciones. Para obtener más información sobre la autorización con Verified Permissions, consulte Autorización con Amazon Verified Permissions.

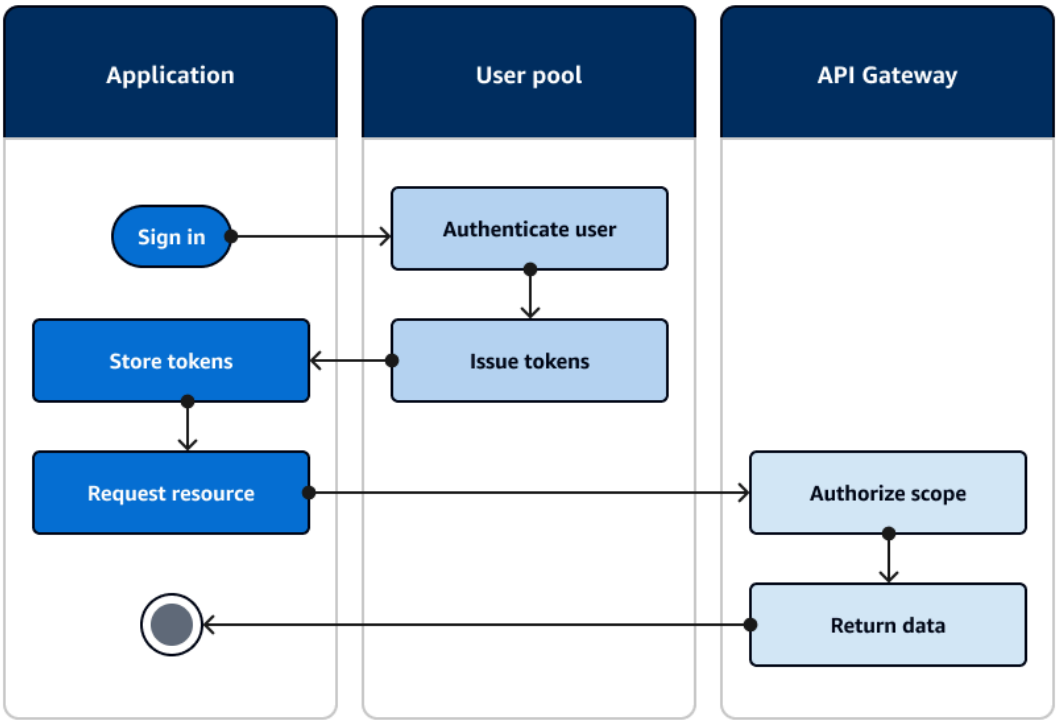
Las solicitudes de autorización de permisos verificados requieren credenciales AWS . Puede implementar algunas de las siguientes técnicas para aplicar las credenciales de forma segura a las solicitudes de autorización.

- Utilice una aplicación web que pueda almacenar secretos en el back-end del servidor.
- Adquirir credenciales autenticadas del grupo de identidades.
- Utilizar el proxy para las solicitudes de los usuarios a través de una API autorizada por un token de acceso y añadir las credenciales de AWS a la solicitud.

Acceso a recursos con API Gateway después del inicio de sesión

Los tokens de los grupos de usuarios de Amazon Cognito suelen utilizarse para autorizar las solicitudes a una API de REST de API Gateway. Los ámbitos de OAuth 2.0 de los tokens de acceso pueden autorizar un método y una ruta, por ejemplo, `HTTP GET` para `/app_assets`. Los identificadores pueden servir como autenticación genérica en una API y pueden transferir los atributos del usuario al servicio de back-end. API Gateway tiene opciones de autorización personalizadas adicionales, como los autorizadores de JWT para las API de HTTP y los autorizadores de Lambda que pueden aplicar una lógica más detallada.

En el siguiente diagrama se ilustra el acceso de una aplicación a una API de REST con los ámbitos de OAuth 2.0 en un token de acceso.



La aplicación debe recopilar los tokens de las sesiones autenticadas y añadirlos como tokens de portadores a un encabezado `Authorization` de la solicitud. Configure el autorizador que ha configurado para la API, la ruta y el método para evaluar el contenido de los tokens. API Gateway devuelve datos solo si la solicitud cumple las condiciones configuradas para el autorizador.

A continuación mostramos varias formas que API Gateway puede utilizar para aprobar el acceso desde una aplicación:

- El token de acceso es válido, no ha caducado y contiene el ámbito correcto de OAuth 2.0. El autorizador de grupos de usuarios de Amazon Cognito para una API de REST es una implementación común con una barrera de entrada baja. También puede evaluar el cuerpo, los parámetros de cadena de consulta y los encabezados de una solicitud a este tipo de autorizador.
- El token de ID 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, una notificación, un atributo o un rol de un token de acceso o ID cumple los requisitos que se definen en una función de Lambda. Un autorizador de Lambda analiza el token del encabezado de la solicitud y lo evalúa para tomar una decisión de autorización. Puede crear una lógica personalizada de constructo en la función o realizar una solicitud de API a Amazon Verified Permissions.

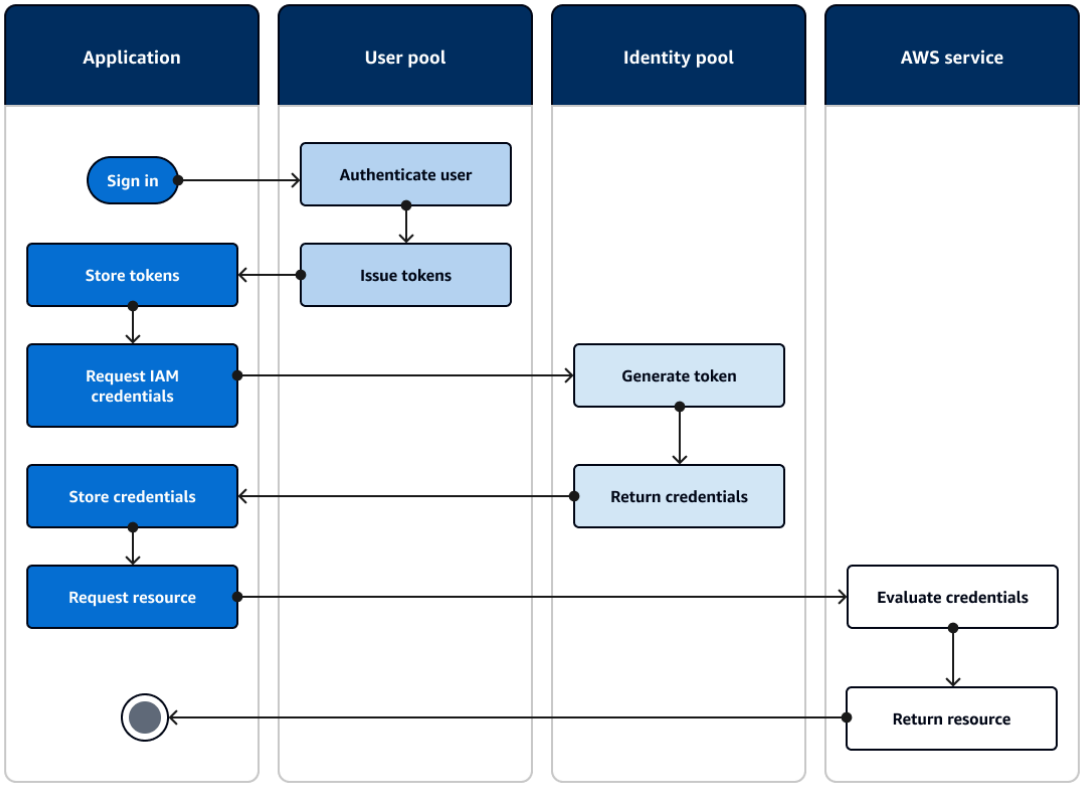
También puede autorizar solicitudes a una API GraphQL de AWS AppSync con tokens de un grupo de usuarios.

Acceso Servicios de AWS usar 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 desde un grupo de usuarios. Al configurar su grupo de usuarios como 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.

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 recurso desde un Servicio de AWS.



Puede usar las credenciales del grupo de identidades para:

- Realizar solicitudes de autorización detalladas a Amazon Verified Permissions con las propias credenciales del 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 back-end de base de datos, como Amazon DynamoDB o Amazon RDS, que autorice las conexiones con IAM.
- Recuperar recursos de aplicaciones desde un bucket de Amazon S3.
- Inicie una sesión con un escritorio WorkSpaces virtual de Amazon.

Los grupos de identidades no funcionan exclusivamente en una sesión autenticada con un grupo de usuarios. También aceptan la autenticación directamente de proveedores de identidades 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 yUso del control de acceso basado en roles. Además, para obtener más información sobre

los grupos de identidades AWS Identity and Access Management, consulteFlujo de autenticación de grupos de identidades.

Configuración de un grupo de usuarios con Consola de administración de AWS

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 Creación de una nueva aplicación en la consola de Amazon Cognito.

Configurar un grupo de identidades con Consola de administración de AWS

El siguiente procedimiento describe cómo usarlo Consola de administración de AWS 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 dar a los usuarios de ese IdP el Rol predeterminado que ha configurado al configurar el 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 función 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 provoca 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 ha seleccionado Elegir rol con solicitud preferred_role en tokens, Amazon Cognito emite credenciales para el rol en la notificación de `cognito:preferred_role` de su usuario. 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 depende del ID del grupo de usuarios de Amazon Cognito. Tiene la estructura siguiente:

```
cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>
```

Puede derivar el valor de la ID ***<region>*** del grupo de usuarios. 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

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";

var cognitoUser = userPool.getCurrentUser();

if (cognitoUser != null) {
```

```
 cognitoUser.getSession(function(err, result) {
  if (result) {
   console.log('You are now logged in.');

   // Create a credentials provider with the user's ID token in the logins map.
   const region = '<region>';
   const credentials = fromCognitoIdentityPool({
    identityPoolId: 'YOUR_IDENTITY_POOL_ID',
    logins: {
     'cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>':
 result.getIdToken().getJwtToken()
    },
    clientConfig: { region }
   });

   // Attach the provider to each service client that needs credentials.
   const client = new SomeServiceClient({ region, credentials });
  }
 });
}
```

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)
```

Descripción de los tokens web JSON (JWT) de los grupos de usuarios

Los tokens son artefactos de autenticación que sus aplicaciones pueden usar como prueba de la autenticación OIDC y para solicitar acceso a los recursos. 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 El punto de conexión 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 base64url. Puede descodificar cualquier ID de Amazon Cognito o token de acceso de `base64url` 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 notificaciones 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

- Descripción del token de identidad (ID)
- Descripción del token de acceso
- Analice los tokens del grupo de usuarios de forma segura
- Tokens de actualización
- Finalización de las sesiones de usuario con la revocación del token
- Verificación de tokens web JSON

- Administración de la caducidad y el almacenamiento en caché de los tokens del grupo de usuarios
- Ámbitos, M2M y servidores de recursos

Descripción del token de identidad (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 tokens web JSON.

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.

El token de identificación incluye las `amr` afirmaciones `acr` y, que describen el nivel de autenticación que cumplió el usuario y los métodos que completó. Para obtener más información, consulte Declaraciones sobre los tokens ACR y AMR.

> **Important**
>
> Cuando el usuario inicia sesión con un inicio de sesión administrado, Amazon Cognito establece cookies de sesión válidas durante 1 hora. Si utiliza el inicio de sesión administrado para la autenticación en su aplicación 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.

Cabecera del token de identificación

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 utilizan un algoritmo criptográfico RS256, que es una firma RSA. 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 acerca de las notificaciones estándar de OpenID Connect (OIDC), consulte la lista OIDC standard claims. Puede añadir notificaciones de diseño propio con un Desencadenador de Lambda anterior a la generación del token.

Esta carga útil es representativa, no exhaustiva. El conjunto de afirmaciones de un token de Amazon Cognito ID aumenta con el tiempo a medida que las nuevas funciones añaden nuevas afirmaciones, por lo que las afirmaciones que contienen tus fichas pueden diferir de las de este ejemplo. Trate este ejemplo como una instantánea de las reclamaciones habituales en lugar de una lista completa y fija, y léalo ¿Cómo analizar este token de forma segura antes de escribir código que analice los tokens de Amazon Cognito.

```
<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

El 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.

> **Important**
>
> Amazon Cognito genera `sub` en un formato de Amazon que no se ajusta a un Cognito-specific formato de UUID específico, incluido el UUID de RFC. No debe validar estrictamente el formato de. `sub` De forma más general, tal y como se describe en la guía sobre cómo analizar de forma segura las reclamaciones relacionadas con el token,

lee cada reclamación según el tipo que se documenta en esta página y no impongas un formato más estricto que el que se indica en la documentación.

`cognito:groups`

Una matriz de cadenas. Cada cadena es el nombre de un grupo de usuarios del que forma parte su usuario. 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. Esta afirmación es siempre una matriz, incluso cuando el usuario pertenece a un solo grupo.

`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`

Una sola cadena. El emisor del token. Esta afirmación identifica el grupo de usuarios que generó el token. La aplicación debe validar que este valor coincide con la URL del emisor prevista para el grupo de usuarios. La reclamación tiene el formato siguiente:

`https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>`

Su grupo de usuarios puede usar un emisor original o actualizado. Los emisores actualizados alojan el mismo contenido de JWKS en varias regiones, lo que mejora la resiliencia y la eficiencia. Para obtener más información, consulte Grupos de usuarios de Amazon Cognito como emisor de OIDC.

`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 de cadenas. Cada cadena es el ARN de un rol de IAM asociado a uno de los grupos de usuarios. Cada grupo de usuarios puede tener un rol de IAM asociado. Esta matriz representa todas las funciones de IAM de los grupos de usuarios, independientemente de su prioridad, y siempre es una matriz incluso cuando contiene una sola función. Para obtener más información, consulte Agregar grupos a un grupo de usuarios.

`aud`

Una sola cadena. 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`

Una matriz de objetos . Cada objeto describe un perfil de proveedor de identidad externo que has vinculado al usuario, ya sea mediante un inicio de sesión federado o vinculando un usuario federado a un perfil local. Cada objeto contiene los campos `userId` (el identificador único del usuario en el proveedor),, `providerName` `providerTypeissuer`, `primary` y. `dateCreated` Todos estos campos son cadenas, con dos detalles de serialización que deben analizarse cuidadosamente: `primary` es la cadena `"true"` o `"false"` no un booleano de JSON, y `dateCreated` es una cadena que contiene el tiempo de Unix en milisegundos (por ejemplo`"1642699117273"`), no un número. El `issuer` campo puede ser. `null` Esta afirmación es siempre una matriz, incluso cuando el usuario tiene una identidad vinculada.

`token_use`

El objetivo para el que se creó el token. En un token de identificación, su valor es `id`.

`auth_time`

Un número de . La hora de autenticación, en formato de hora Unix, cuando el usuario completó la autenticación.

`exp`

Un número de . La hora de caducidad, en formato de hora Unix, en la que caduca el token de tu usuario.

`iat`

Un número de . El emitido en el momento, en formato de hora de Unix, cuando Amazon Cognito emitió el token de su usuario.

`jti`

El identificador único del JWT.

`email`

Una sola cadena. La dirección de correo electrónico del usuario, cuando el `email` atributo esté presente y pueda leerlo el cliente de la aplicación.

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 de los grupos de usuarios siempre llevan el prefijo `custom:`.

¿Cómo analizar este token de forma segura

La afirmación establecida en un token de Amazon Cognito ID no es fija. Las afirmaciones que contiene un token varían según el flujo de autenticación, la pertenencia al grupo del usuario y el proveedor de identidad. También dependen de las funciones que estén habilitadas en el grupo de usuarios. La especificación OpenID Connect solo requiere`iss`,, y `sub` `aud` `expiat`, y están presentes en todos los tokens de Amazon Cognito ID. Amazon Cognito también añade `token_use`

siempre el valor`id`, que debe comprobar para confirmar que ha recibido un token de identificación en lugar de un token de acceso. Trate todas las demás solicitudes como opcionales y supeditadas a la forma en que el usuario se haya autenticado.

El token de ID es uno de los varios tipos de token de Amazon Cognito y se aplican las mismas reglas de análisis a todos ellos. Para obtener más información sobre cómo analizar los tokens de forma segura, consulte. Analice los tokens del grupo de usuarios de forma segura

Ejemplos de tokens de ID por escenario

Carga útil predeterminada del token de IDCombina muchos reclamos opcionales en un solo ejemplo. En la práctica, los reclamos que conlleva un token dependen de la forma en que el usuario haya iniciado sesión. En los siguientes ejemplos, se muestran las combinaciones de notificaciones que Amazon Cognito emite en cuatro escenarios comunes. Cada ejemplo es una carga útil decodificada, no un JWT firmado y serializado: se trata de un token real`header.payload.signature`, codificado en base64url y con la clave de su grupo de usuarios. RS256-signed Los valores del ID del grupo de usuarios `sub` y de la marca de tiempo son marcadores de posición ilustrativos.

Símbolo de identificación de código de autorización básico

El token de identidad que consumen la mayoría de las aplicaciones después de intercambiar un código de autorización en el punto final del token. El `token_use` valor de lo `id` distingue de un token de acceso. La `nonce` afirmación se debe a que el cliente envió un `nonce` parámetro en la solicitud de autorización para evitar los ataques de repetición.

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "my-test-user",
    "email": "my-test-user@example.com",
    "email_verified": true,
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888",
    "nonce": "n-0S6_WzA2Mj"
}
```

Pertenencia a grupos y roles

Amazon Cognito añade las notificaciones de autorización de grupos y roles cuando el usuario pertenece a uno o más grupos de grupos de usuarios. Las `cognito:roles` afirmaciones `cognito:groups` y son matrices; léelas como matrices aunque contengan un solo elemento. `cognito:preferred_role`es una cadena única.

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "my-test-user",
    "cognito:groups": ["admins", "beta-testers"],
    "cognito:roles": ["arn:aws:iam::111122223333:role/my-test-role"],
    "cognito:preferred_role": "arn:aws:iam::111122223333:role/my-test-role",
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888"
}
```

Inicio de sesión federado

El usuario que haya iniciado sesión a través de un proveedor de identidad externo, como una red social o un SAML/OIDC proveedor, lleva la `identities` afirmación, que describe el enlace del proveedor externo.

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "LoginWithAmazon_amzn1.account.EXAMPLE",
    "identities": [
        {
            "userId": "amzn1.account.EXAMPLE",
            "providerName": "LoginWithAmazon",
            "providerType": "LoginWithAmazon",
            "issuer": null,
            "primary": "true",
            "dateCreated": "1642699117273"
```

```
        }
    ],
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888"
}
```

> **Note**
>
> Están dentro de cada `identities` objeto `primary` y `dateCreated` se serializan como cadenas (`"true"`y`"1642699117273"`), no como un booleano y un número JSON. Una comparación como, por ejemplo, se evalúa `identity.primary === true` silenciosamente como falsa. Se lee `primary` como la cadena `"true"` o `"false"` y `dateCreated` como una cadena que contiene el tiempo de Unix en milisegundos.

Fuerza de autenticación (ACR y AMR)

Cuando su grupo de usuarios informa del nivel de autenticación alcanzado por el usuario, Amazon Cognito añade las notificaciones `acr` y`amr`. El siguiente token proviene de un usuario que se autenticó con una contraseña y un TOTP desde una aplicación de autenticación, que es el nivel más alto. Cada token contiene exactamente uno de los niveles posibles. `acr` Para ver el conjunto fijo de niveles, las combinaciones de factores que satisfacen cada uno y la lista completa de `amr` valores, consulte[Niveles de autenticación con afirmaciones de ACR y AMR](#).

```
{
    "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
    "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
    "aud": "6o7h8i9jexampleclientid",
    "cognito:username": "my-test-user",
    "acr": "urn:cognito:loa:4",
    "amr": ["pwd", "otp", "mfa"],
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "token_use": "id",
    "auth_time": 1759345200,
    "iat": 1759345200,
    "exp": 1759348800,
    "jti": "11112222-3333-4444-5555-666677778888"
```

```
}
```

Para solicitar un nivel de autenticación se requiere el plan de funciones Essentials o Plus. Amazon Cognito calcula `amr` las reclamaciones `acr` y no las puede modificar un activador Lambda previo a la generación del token. La aplicación es responsable de traducir los niveles de Amazon Cognito a otro estándar, como el NIST o el eIDAS.

Identificador, token, firma

La firma del token de ID se calcula en función del encabezado y la carga útil 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 Verificar un token web JSON. Verificación de tokens web JSON.

Descripción 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 REST con información de su grupo de usuarios. Para obtener más información, consulte Controlar el acceso a las API de REST mediante grupos de usuarios de Amazon Cognito como autorizador. También puede usar Amazon Cognito como autorizador de JWT para una API HTTP. Para obtener más información, consulte Controlar el acceso a las API HTTP con autorizadores JWT en API Gateway. Para generar un token de acceso con ámbitos personalizados, debe solicitarlo a través de los puntos de conexión públicos de su grupo de usuarios.

Con el plan de características Essentials o Plus, también puede implementar un desencadenador de Lambda Antes de la generación del token que añada ámbitos a los tokens de acceso en tiempo de ejecución. Para obtener más información, consulte Desencadenador de Lambda anterior a la generación del token.

El token de acceso de un usuario con el ámbito `openid` es un permiso para solicitar más información sobre los atributos de su usuario a El punto de conexión userInfo. La cantidad de información del punto de conexión `userInfo` se deriva de los ámbitos adicionales del token de acceso: por ejemplo, `profile` para todos los datos del usuario y `email` para su dirección de correo electrónico.

El token de acceso de un usuario con el ámbito `aws.cognito.signin.user.admin` es el permiso para leer y escribir los atributos del usuario, enumerar los factores de autenticación, configurar las preferencias de autenticación multifactor (MFA) y administrar los dispositivos recordados. El nivel de acceso a los atributos que tu token de acceso otorga a este ámbito coincide con los read/write permisos de atributos que asignes a tu cliente de aplicación.

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 tokens web JSON. 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 el inicio de sesión administrado. El inicio de sesión administrado establece cookies del navegador que son válidas durante una hora. Si configura una duración del token de acceso inferior a una hora, esto no afecta a la validez de la cookie de inicio de sesión administrado ni a la capacidad de los usuarios para volver a autenticarse sin credenciales adicionales durante una hora después del inicio de sesión inicial.

El token de acceso contiene las notificaciones de forma aditiva, del mismo modo que el token de ID, y se aplican las mismas reglas de análisis. Para obtener más información sobre cómo analizar los tokens de forma segura, consulte. Analice los tokens del grupo de usuarios de forma segura

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 utilizan 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 notificaciones de diseño propio con un Desencadenador 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",
   "aud": "https://api.example.com",
```

```
   "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`

El 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.

> **Important**
>
> Amazon Cognito genera `sub` en un formato de Amazon que no se ajusta a un Cognito-specific formato de UUID específico, incluido el UUID de RFC. No debe validar estrictamente el formato de. `sub`

`cognito:groups`

Una matriz con los nombres de los grupos de usuarios que tienen a su usuario como miembro.

`iss`

El emisor del token. Esta afirmación identifica el grupo de usuarios que generó el token. La aplicación debe validar que este valor coincide con la URL del emisor prevista para el grupo de usuarios. La reclamación tiene el formato siguiente:

`https://cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE`

Su grupo de usuarios puede usar un emisor original o actualizado. Los emisores actualizados alojan el mismo contenido de JWKS en varias regiones, lo que mejora la resiliencia y la eficiencia. Para obtener más información, consulte Grupos de usuarios de Amazon Cognito como emisor de OIDC.

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.

aud

La URL de la API que se pretende autorizar con el token de acceso. Está presente solo si la aplicación solicitó una vinculación de recursos a su servidor de autorización.

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 deja de validar 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 emitidos al usuario que ha iniciado sesión. Los ámbitos definen el acceso que el token proporciona a las API externas, las operaciones de autoservicio del usuario y los datos del usuario en el punto de conexión `userInfo`. 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.

Más recursos

- How to customize access tokens in Amazon Cognito user pools

Machine-to-machine carga útil del token de acceso

Como un token de acceso de máquina a máquina (M2M) autoriza una aplicación en lugar de a un usuario, su carga útil es diferente de la del token de acceso de usuario descrito anteriormente. En la autorización M2M, la notificación `sub` contiene el ID del cliente de la aplicación que solicitó el token, no un identificador de usuario, y las notificaciones y tienen el mismo valor`sub`. `client_id` El token incluye los ámbitos personalizados que autorizan las solicitudes a tu servidor de recursos, junto con afirmaciones estándar como`iss`, `token_use` (con un valor de`access`), `expiat`, y. `jti`

Como la autorización M2M no involucra a ningún usuario y no emite ningún token de actualización, un token de acceso M2M omite varias afirmaciones que contiene un token de acceso de usuario. Su servidor de recursos no puede esperar estas afirmaciones:

- Las `cognito:groups` afirmaciones `username` y, que identifican a un usuario autenticado y a su pertenencia a un grupo.
- La `device_key` afirmación, que identifica un dispositivo de usuario recordado.
- La `origin_jti` afirmación, que vincula un token a un token de actualización revocable que la autorización M2M no emite.

Firma del token de acceso

La firma del token de acceso, firmada con la clave anunciada en el punto de conexión `.well-known/jwks.json`, valida la integridad del encabezado y la carga útil del token. Cuando utilice los tokens de acceso para autorizar el acceso a las API externas, configure siempre el autorizador de la API para que verifique esta firma con la clave que la firmó. Para obtener más información, consulte Verificación de tokens web JSON.

Analice los tokens del grupo de usuarios de forma segura

El conjunto de reclamos de un token de Amazon Cognito no es fijo. Las afirmaciones que contiene su token varían según el flujo de autenticación, la pertenencia al grupo del usuario y el proveedor de identidad, así como según las funciones que habilite en su grupo de usuarios. Amazon Cognito añade nuevas afirmaciones de forma aditiva a lo largo del tiempo: introduce nuevas claves, pero no cambia el nombre, el tipo ni elimina las afirmaciones existentes que se documentan en esta guía. Estas reglas se aplican tanto a los tokens de identificación como a los de acceso.

No escribas tu propio analizador de fichas. Usa una biblioteca de tokens web JSON (JWT) y un analizador JSON que sigan funcionando a medida que el token crece. Las dos reglas que impiden prácticamente cualquier irrupción en el mundo real son las dos primeras que se indican a continuación: tolere las afirmaciones que no reconozca y haga suposiciones correctas sobre el tipo y la cardinalidad de cada afirmación que lea.

Ignora las afirmaciones que no reconozcas

Lee las afirmaciones que necesita tu solicitud y olvídate del resto. No rechace, rechace ni arroje un error cuando un token contenga una afirmación que no figure en esta guía. Las nuevas afirmaciones se añaden de forma aditiva, y un token que contenga una afirmación que no figure en la lista sigue siendo válida.

Lee cada afirmación en su tipo documentado

Un analizador de JSON identifica el tipo primitivo de cada valor a partir del propio JSON, por lo que el riesgo está en las suposiciones que el código haga sobre ese valor. Trata cada afirmación según el tipo y la cardinalidad que se documenta en esta guía. Una afirmación documentada como matriz es siempre una matriz, incluso cuando actualmente contiene un único valor. No lo contraigas en forma de cadena. Por ejemplo, `cognito:groups`cognito:roles, y `identities` son siempre matrices. El mismo principio se aplica a las reclamaciones definidas en otras especificaciones. Por ejemplo, cuando está presente`amr`, es una matriz de cadenas tal como se define en el RFC 8176 del sitio web de la IETF, por lo que debe leerse en ese tipo de caracteres.

No sobrevalides los formatos de las reclamaciones

Valide lo que exija la especificación de una reclamación y nada más. Por ejemplo, `sub` es una cadena única, pero no se garantiza que esté en ningún formato de UUID en particular. Algunos valores incluidos en las notificaciones personalizadas, por ejemplo, `identities` se serializan como cadenas en lugar de como números o booleanos JSON, y no se garantiza que los valores

opacos tengan una longitud determinada. Si se impone un formato más estricto que el establecido en la documentación, se produce un error en los tokens válidos.

No asuma el número de reclamaciones ni su orden

Las reclamaciones de una ficha pueden aparecer en cualquier orden y el recuento puede cambiar de una ficha a otra y a lo largo del tiempo. Diríjase a cada afirmación por su nombre en lugar de por su posición, y analice el token como un objeto JSON general y no como una estructura fija.

Haga un presupuesto para aumentar el tamaño del token

A medida que el conjunto de reclamos crece, también crece el tamaño total del token. No asuma un tamaño de token fijo o máximo en los búferes, anchos de columna, encabezados o cookies.

El documento de `openid-configuration` descubrimiento sigue el mismo enfoque aditivo y compatible con versiones futuras. Para obtener más información sobre cómo analizar ese documento de forma segura, consulte. Prepárese para los cambios en el documento de descubrimiento

Tokens 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.

Obtener nuevos tokens de acceso e identidad con un token de actualización

Amazon Cognito emite tokens de actualización en respuesta a una autenticación correcta con el flujo de códigos de autorización de inicio de sesión administrado y con operaciones de API o métodos de SDK. El token de actualización devuelve nuevos tokens de ID y acceso y, si lo desea, un nuevo token de actualización. Puede utilizar tokens de actualización de las siguientes maneras.

GetTokensFromRefreshToken

La operación de la GetTokensFromRefreshTokenAPI emite nuevos identificadores y identificadores de acceso a partir de un token de actualización válido. También obtendrá un nuevo token de actualización si ha activado la rotación del token de actualización.

InitiateAuth y AdminInitiateAuth

Las AdminInitiateAuthoperaciones de la InitiateAuthAPI incluyen el flujo `REFRESH_TOKEN_AUTH` de autenticación. En este flujo, se pasa un token de actualización y se obtienen tokens de ID y de acceso nuevos. No puede autenticarse con `REFRESH_TOKEN_AUTH` en los clientes de aplicación que tengan habilitada la rotación del token de actualización.

Punto de conexión de token OAuth

El punto de conexión del token de los grupos de usuarios con un dominio tiene un tipo de concesión `refresh_token` que emite nuevos tokens de ID, acceso y, si lo desea (con la rotación del token de actualización), de actualización, a partir de un token de actualización válido.

Rotación de tokens de actualización

Si lo desea, puede configurar la rotación del token de actualización en el cliente de su aplicación. Con la rotación del token de actualización, su cliente puede invalidar el token de actualización original y emitir un nuevo token de actualización con cada actualización del token. Cuando esta configuración está habilitada, cada solicitud correcta en todas las formas de actualización del token devuelve un identificador, un acceso y un token de actualización nuevos. Cuando esta configuración está deshabilitada, las solicitudes de actualización de los tokens solo devuelven nuevos tokens de acceso e ID, y el token de actualización original sigue siendo válido. El nuevo token de actualización es válido durante el tiempo restante del token de actualización original. Puede configurar los clientes de aplicación para que roten los tokens de actualización o para que transfieran el token de actualización original. Para permitir los reintentos durante un período breve, también puede configurar un período de gracia para el token de actualización original de hasta 60 segundos.

> **Note**
>
> Rotation vuelve a emitir un token de actualización dentro del mismo período de validez fijo. No amplía la ventana. Cada nuevo token de actualización es válido solo durante el tiempo restante del `RefreshTokenValidity` período original, sin importar cuántas veces gire el token. Por ejemplo, supongamos que lo `RefreshTokenValidity` estableces en 30 días y una sesión rota su token de actualización todos los días. La sesión sigue caducando 30 días después del primer inicio de sesión, no 30 días después de la rotación más reciente. Para mantener la sesión de un usuario pasada esa ventana, debe volver a autenticarse.

Cosas que debe saber acerca de la rotación de tokens de actualización

- Después de habilitar la rotación de tokens de actualización, se agregan nuevas reclamaciones en tokens web JSON desde su grupo de usuarios. Las notificaciones `origin_jti` y `jti` se agregan a los tokens de acceso e ID. Estas afirmaciones aumentan el tamaño de los JWT.
- La rotación del token de actualización no es compatible con el flujo de autenticación `REFRESH_TOKEN_AUTH`. Para implementar la rotación del token de actualización, debes deshabilitar este flujo de autenticación en el cliente de la aplicación y diseñar la aplicación para que envíe solicitudes de actualización del token con la operación de GetTokensFromRefreshTokenAPI o el método del SDK equivalente.
- Si la rotación de los tokens de actualización está inactiva, puede completar las solicitudes de actualización de los tokens con `GetTokensFromRefreshToken` o `REFRESH_TOKEN_AUTH`.
- Cuando la función para recordar dispositivos está activa en su grupo de usuarios, debe proporcionar la clave del dispositivo en las solicitudes `GetTokensFromRefreshToken`. Si su usuario no tiene una clave de dispositivo confirmada que la aplicación envíe en la solicitud de autenticación inicial, Amazon Cognito emitirá una nueva. Para actualizar los tokens en esta configuración, debe proporcionar una clave de dispositivo, tanto si especificó una en `AuthParameters` como si recibió una nueva en la respuesta de autenticación.
- Puede pasar `ClientMetadata` al desencadenador de Lambda Antes de la generación del token en su solicitud `GetTokensFromRefreshToken`. Estos datos, que se transfieren al evento de entrada del desencadenador, proporcionan un contexto adicional que puede utilizar en la lógica personalizada de la función de Lambda.

Como práctica recomendada de seguridad, habilite la rotación de tokens de actualización en los clientes de sus aplicaciones.

Enable refresh token rotation (console)

El siguiente procedimiento activa o desactiva la rotación del token de actualización para el cliente de aplicación. Este procedimiento requiere un cliente de aplicación existente. Para obtener más información sobre cómo crear un cliente de aplicación, consulte Application-specific configuración con clientes de aplicaciones.

Cómo habilitar la rotación de tokens de actualización

1. Vaya a la consola de Amazon Cognito. Si se te solicita, introduce tus credenciales. AWS
2. Elija User Pools (Grupos de usuarios).

3. Elija en la lista un usuario existente.
4. Navegue hasta el menú Clientes de aplicación y seleccione un cliente de aplicación existente.
5. Seleccione Editar en la sección de Información del cliente de aplicación de la página.
6. En Configuraciones de seguridad avanzadas, busque la opción Habilitar la rotación de tokens de actualización.
7. Para habilitar la rotación, marque la casilla de verificación. Para deshabilitar la rotación, quite la selección en la casilla de verificación.
8. En Período de gracia de rotación de tokens de actualización, introduzca el número de segundos, hasta 60, que desee establecer como retraso antes de que se revoque el token de actualización rotado.

Enable refresh token rotation (API)

Configura la rotación del token de actualización en una solicitud de UpdateUserPoolClientAPI CreateUserPoolClienteo en una solicitud. El siguiente cuerpo parcial de la solicitud activa la rotación del token de actualización y establece el período de gracia en 10 segundos.

```
"RefreshTokenRotation" : {
   "Feature" : "ENABLED",
   "RetryGracePeriodSeconds" : 10
}
```

Actualización de los tokens de API y SDK

Hay dos formas de usar el token de actualización para obtener un nuevo ID y acceder a los tokens con la API de grupos de usuarios, en función de si la rotación del token de actualización está activa o no. En los clientes de aplicaciones con la rotación del token de actualización activa, usa la operación de GetTokensFromRefreshTokenAPI. En los clientes de aplicaciones sin rotación del token de actualización, usa el `REFRESH_TOKEN_AUTH` flujo de las AdminInitiateAuthoperaciones de la InitiateAuthAPI.

> **Note**
>
> Los usuarios pueden autenticarse con grupos de usuarios en el inicio de sesión gestionado o en aplicaciones personalizadas que usted cree con AWS los SDK y

las operaciones de la API de Amazon Cognito. El flujo `REFRESH_TOKEN_AUTH` y `GetTokensFromRefreshToken` pueden completar la actualización del token para los usuarios con inicio de sesión administrado. La actualización de los tokens en las aplicaciones personalizadas no afecta a las sesiones de inicio de sesión administrado. Estas sesiones se configuran en una cookie del navegador y son válidas durante una hora. La respuesta a `GetTokensFromRefreshToken` emite tokens de ID, de acceso y, opcionalmente, de actualización, pero no renueva la cookie de sesión del inicio de sesión administrado. `REFRESH_TOKEN_AUTH` no está disponible en los clientes de aplicación con la rotación de los tokens de actualización habilitada.

GetTokensFromRefreshToken

GetTokensFromRefreshTokendevuelve los nuevos identificadores, de acceso y de actualización de una solicitud que usted autorice con un token de actualización. A continuación, se muestra un ejemplo de cuerpo de solicitud para `GetTokensFromRefreshToken`. Puede enviar los metadatos del cliente a los desencadenadores de Lambda en las solicitudes de esta operación.

```
{
    "RefreshToken": "eyJjd123abcEXAMPLE",
    "ClientId": "1example23456789",
    "ClientSecret": "myappclientsecret123abc",
    "ClientMetadata": {
      "MyMetadataKey" : "MyMetadataValue"
   },
}
```

AdminInitiateAuth/InitiateAuth

Para usar el token de actualización cuando la rotación del token de actualización esté inactiva, usa las operaciones AdminInitiateAutho InitiateAuthAPI. 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.

A continuación, se muestra un ejemplo del cuerpo de una solicitud para una actualización de un token con la API `InitiateAuth` o `AdminInitiateAuth`.

```
{
```

```
    "AuthFlow": "REFRESH_TOKEN_AUTH",
    "ClientId": "1example23456789",
    "UserPoolId": "us-west-2_EXAMPLE",
    "AuthParameters": {
        "REFRESH_TOKEN": "eyJjd123abcEXAMPLE",
        "SECRET_HASH": "kT5acwCVrbD6JexhW3EQwnRSe6fLuPTRkEQ50athqv8="
    }
}
```

Actualización del token de OAuth

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.

Las solicitudes al punto de conexión del token están disponibles en los clientes de aplicación con la rotación del token de actualización activa y en aquellos en los que está inactiva. Cuando la rotación del token de actualización está activa, el punto de conexión del token devuelve un nuevo token de actualización.

A continuación, se muestra un ejemplo de solicitud con un token de actualización.

```
POST /oauth2/token HTTP/1.1
Host: auth.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw
Content-Length: **

client_id=1example23456789&grant_type=refresh_token&refresh_token=eyJjd123abcEXAMPLE
```

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 Finalización de las sesiones de usuario con la revocación del token.

> **Note**
>
> Al revocar el token de actualización, se revocan todos los tokens de ID y acceso que Amazon Cognito emitió a partir de las solicitudes de actualización con ese token.

Para cerrar la sesión de los usuarios en todas las sesiones en las que hayan iniciado sesión actualmente, revoque todos sus tokens o solicitudes de API. GlobalSignOutAdminUserGlobalSignOut 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 del inicio de sesión administrado 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 con el inicio de sesión administrado, rediríjalos a Punto de conexión Logout, donde Amazon Cognito borra su 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 algunas aplicaciones en las que han mantenido sesión iniciada con sus tokens de actualización. Para cerrar la sesión del usuario para una sola sesión, revoque el token de actualización. Cuando tu usuario quiera cerrar sesión en todas las sesiones autenticadas, genera una solicitud de API. GlobalSignOut Tu aplicación puede ofrecer a tu usuario una opción como Cerrar sesión en todos los dispositivos. `GlobalSignOut`acepta el token de acceso válido de un usuario (inalterado, no caducado ni revocado). 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, consulteLista de operaciones de API agrupadas por modelo de autorización.

Finalización de las sesiones de usuario con la revocación del token

Puede revocar los tokens de actualización y las sesiones de los usuarios finales mediante los siguientes métodos. 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.

RevokeToken operación

RevokeTokenrevoca todos los tokens de acceso de un token de actualización determinado, incluido el token de acceso inicial al iniciar sesión de forma interactiva. Esta operación no afecta a ninguno de los demás tokens de actualización del usuario ni a los símbolos de ID y acceso secundarios de esos otros tokens de actualización.

Punto de conexión de revocación

El punto de conexión de revocación revoca un token de actualización determinado y todos los identificadores de acceso e ID que generó el token de actualización. Este punto de conexión también revoca el token de acceso inicial del inicio de sesión interactivo. Las solicitudes a este punto de conexión no afectan a ninguno de los demás tokens de actualización del usuario ni a los símbolos de ID y acceso secundarios de esos otros tokens de actualización.

GlobalSignOut operación

GlobalSignOutes una operación de autoservicio que un usuario autoriza con su token de acceso. Esta operación revoca todos los tokens de actualización, ID y acceso del usuario solicitante.

AdminUserGlobalSignOut operación

AdminUserGlobalSignOutes una operación del lado del servidor que un administrador autoriza con credenciales de IAM. Esta operación revoca todos los tokens de actualización, ID y acceso del usuario de destino.

Factores que debe tener en cuenta en la revocación de tokens

- La solicitud para revocar un token de actualización debe incluir el ID del cliente que se utilizó para obtener el token.
- Los JWT de grupos de usuarios 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.

- Cuando se crea un nuevo cliente de grupos de usuarios, la revocación de tokens se habilita de forma predeterminada.
- Solo puede revocar tokens de actualización en clientes de aplicación con la revocación de tokens habilitada.
- Después de habilitar la revocación de tokens, se agregan nuevas notificaciones 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.
- Cuando inhabilita la revocación de tokens en un cliente de aplicación donde estaba habilitada anteriormente, los tokens revocados no vuelven a activarse.
- Cuando deshabilita una cuenta de usuario (lo que revoca los tokens de actualización y acceso), los tokens revocados no se activan si vuelve a habilitar la cuenta de usuario.
- Al crear un nuevo cliente de grupo de usuarios mediante la Consola de administración de AWS, la API o la AWS API AWS CLI, la revocación del token está habilitada 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. Puede habilitar la revocación de tokens 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`.

Para crear o modificar un cliente de aplicación con la revocación de token habilitada con la API de Amazon Cognito o con AWS un SDK, incluya el siguiente parámetro en CreateUserPoolClientsu solicitud de API UpdateUserPoolClienteo en su solicitud.

```
"EnableTokenRevocation": true
```

Para configurar la revocación de tokens en la consola de Amazon Cognito, seleccione un cliente de aplicación en el menú Clientes de aplicación de su grupo de usuarios. Seleccione el botón Editar en

la Información del cliente de aplicación y active o desactive la revocación de tokens en Configuración avanzada.

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.

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 tokens web JSON

Los tokens web JSON (JWT) se pueden descodificar, leer y modificar fácilmente. Un token de acceso modificado crea el riesgo de que se produzca un escalado de privilegios. Un token de ID modificado crea un riesgo de suplantación de identidad. La aplicación confía en el grupo de usuarios como emisor del token, pero ¿qué sucede si un usuario intercepta el token en tránsito? Debe asegurarse de que la aplicación reciba el token que Amazon Cognito ha emitido.

Amazon Cognito emite tokens que utilizan algunas de las características de integridad y confidencialidad de la especificación OpenID Connect (OIDC). Los tokens del grupo de usuarios indican la validez con objetos como la fecha de caducidad, el emisor y la firma digital. La firma, que es el tercer y último segmento del JWT delimitado por `.`, es el componente clave de la validación del token. Un usuario malintencionado puede modificar un token, pero si la aplicación recupera la clave pública y la compara con la firma, no coincidirán. Cualquier aplicación que procese JWT de una autenticación de OIDC debe realizar esta operación de verificación cada vez que inicie sesión.

En esta página, hacemos algunas recomendaciones generales y específicas para la verificación de los JWT. El desarrollo de aplicaciones abarca diversos lenguajes de programación y plataformas. Dado que Amazon Cognito implementa OIDC lo suficientemente cerca de la especificación pública, cualquier biblioteca de JWT acreditada del entorno de desarrollador que elija puede gestionar los requisitos de verificación.

Estos pasos describen cómo verificar un JSON Web Token (JWT) de grupo de usuarios.

Temas

- Requisitos previos
- Validación de 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 token web JSON (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.

Validación de los tokens con aws-jwt-verify

En una Node.js aplicación, 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 Node.js aplicación o en un AWS Lambda autorizador, consulta esta aws-jwt-verifypágina GitHub.

> **Note**
>
> Cuando la afirmación `iss` (del emisor) de un token no coincide con el grupo de usuarios `CognitoJwtVerifier` con el que lo configuraste, la verificación falla con un. `ParameterValidationError` Esto suele ocurrir cuando se presenta un token a un verificador de un grupo de usuarios o AWS una región diferentes. Confirme que el `userPoolId` identificador al que accede `CognitoJwtVerifier.create()` coincide con el emisor de los tokens que recibe su solicitud.

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`. `kid` es una referencia truncada a una clave de firma privada RSA de 2048 bits que se encuentra en poder del grupo de usuarios.

Carga útil

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 base64url como el encabezado y la carga útil. 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 útil son JSON y están codificados en base64url. Puede identificarlos por los caracteres de apertura `eyJ` que se descodifican para formar el carácter inicial `{`. Si su usuario presenta un JWT codificado en base64url a su aplicación y no está en el formato `[JSON Header].[JSON Payload].[Signature]`, no es un token de Amazon Cognito válido, por lo que puede descartarlo.

El siguiente ejemplo de aplicación verifica los tokens del grupo de usuarios con `aws-jwt-verify`.

```
// cognito-verify.js
// Usage example: node cognito-verify.js eyJra789ghiEXAMPLE

const { CognitoJwtVerifier } = require('aws-jwt-verify');

// Replace with your Amazon Cognito user pool ID
const userPoolId = 'us-west-2_EXAMPLE';

async function verifyJWT(token) {
  try {
    const verifier = CognitoJwtVerifier.create({
      userPoolId,
      tokenUse: 'access', // or 'id' for ID tokens
      clientId: '1example23456789', // you must verify the token audience
    });

    const payload = await verifier.verify(token);
    console.log('Decoded JWT:', payload);
  } catch (err) {
    console.error('Error verifying JWT:', err);
  }
}

// Example usage
if (process.argv.length < 3) {
  console.error('Please provide a JWT token as an argument.');
  process.exit(1);
}

const MyToken = process.argv[2];
verifyJWT(MyToken);
```

Validación del JWT

La firma JWT es una combinación con hash del encabezado y la carga útil. 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 utilizan 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. Base64urlUInt-encoded

Módulo RSA (**`n`**)

El parámetro `n` contiene el valor del módulo de la clave pública RSA. Se representa como un Base64urlUInt-encoded valor.

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.

Comprobar las notificaciones

Para comprobar las notificaciones JWT

1. Comprueba que el token no haya caducado. Descodifique el token y compare la reclamación `exp` con la hora actual. Como los JWT del grupo de usuarios son independientes y tienen su propia fecha de caducidad, puede comprobar la caducidad sin conexión, sin necesidad de realizar una llamada de red a Amazon Cognito.

   > **Note**
   >
   > Las operaciones de la API, como GetUserel El punto de conexión userInfo punto final, devuelven un error cuando se trata de un token de acceso caducado, pero el rechazo de una solicitud es un efecto secundario de la autorización y no un método de verificación del token. No utilices estas llamadas de red para comprobar la caducidad: decodifica el token y, en su lugar, compara la afirmación. `exp`

2. La `aud` afirmación de un token de ID y la `client_id` afirmación de un token de acceso deben coincidir con el ID de cliente de la aplicación que se creó en el grupo de usuarios de Amazon Cognito.
3. La afirmación del emisor (`iss`) debe coincidir con su grupo de usuarios. Por ejemplo, un grupo de usuarios creado en la `us-east-1` región tiene el siguiente `iss` valor:

   `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`.
5. Confirma que el token no se haya revocado. Es posible que un token con una firma válida y una `exp` reclamación no vencida se haya revocado antes de su fecha de caducidad, por ejemplo, cuando un usuario cierra sesión o cuando llamas a la operación de la API. RevokeToken Las comprobaciones de firma y caducidad fuera de línea no pueden detectar la revocación por sí solas. Si su aplicación debe cumplir con la revocación antes de que caduque un token, verifique el estado del token con Amazon Cognito (por ejemplo, con una solicitud de API autorizada)

o mantenga la vida útil del token de acceso corta. Para obtener más información, consulte Finalización de las sesiones de usuario con la revocación del token.

Ahora puede confiar en las notificaciones del token.

> **Note**
>
> Algunos Servicios de AWS pueden verificar los JWT de Amazon Cognito por usted, en lugar de hacerlo en el código de su aplicación. Si utiliza Amazon Cognito como autorizador de JWT para una API HTTP de Amazon API Gateway, proporciona los mismos valores de emisor y público en la configuración del autorizador. Establezca el emisor en el punto final de su grupo de usuarioshttps://cognito-idp.*<Region>*.amazonaws.com/*<userPoolId>*, que también es el valor del `issuer` campo en el documento de su grupo de usuarios. `/.well-known/openid-configuration` Configura la audiencia con el ID de cliente de tu aplicación.
> API Gateway compara a la audiencia con la `aud` afirmación del token o con la `client_id` afirmación cuando el token no tiene ninguna `aud` reclamación. Los tokens de Amazon Cognito ID se incluyen y los de acceso se llevan `client_id` y no`aud`, a `aud` menos que la solicitud incluya un enlace entre el servidor y el recurso. En el caso de un token de ID o un token de acceso sin vinculación de recursos, defina la audiencia con el ID de cliente de la aplicación. Cuando tu aplicación solicita un token de acceso enlazado a un servidor de recursos, la notificación del `aud` token se establece en el identificador de recursos (la URL del servidor de recursos) en lugar de en el ID de cliente de la aplicación; en ese caso, configura la URL del servidor de recursos como la audiencia para que API Gateway coincida con ella. Para ver los pasos de configuración de API Gateway, consulte Controlar el acceso a las API de REST mediante grupos de usuarios de Amazon Cognito como autorizador y Controlar el acceso a las API HTTP con autorizadores de JWT en API Gateway.
> Un Application Load Balancer también puede autenticar a los usuarios con su grupo de usuarios. El balanceador de cargas transfiere las reclamaciones de los usuarios firmadas a tus destinos en un encabezado JWT-formatted `x-amzn-oidc-data` HTTP que la aplicación verifica. Para obtener más información, consulte Autenticar usuarios mediante un Equilibrador de carga de aplicación.

Administración de la caducidad y el almacenamiento en caché de los tokens del grupo de usuarios

Su aplicación debe completar correctamente una de las siguientes solicitudes cada vez que desee obtener un nuevo token web JSON (JWT).

- Solicite las credenciales o la concesión del código de autorización desde el Punto de conexión de token.
- Solicita un token de acceso con las credenciales del cliente con la operación de la GetClientTokenAPI, que no requiere un dominio de grupo de usuarios.
- Solicite una concesión implícita desde las páginas de inicio de sesión administrado.
- Autentique a un usuario local en una solicitud de API de Amazon Cognito como. InitiateAuth

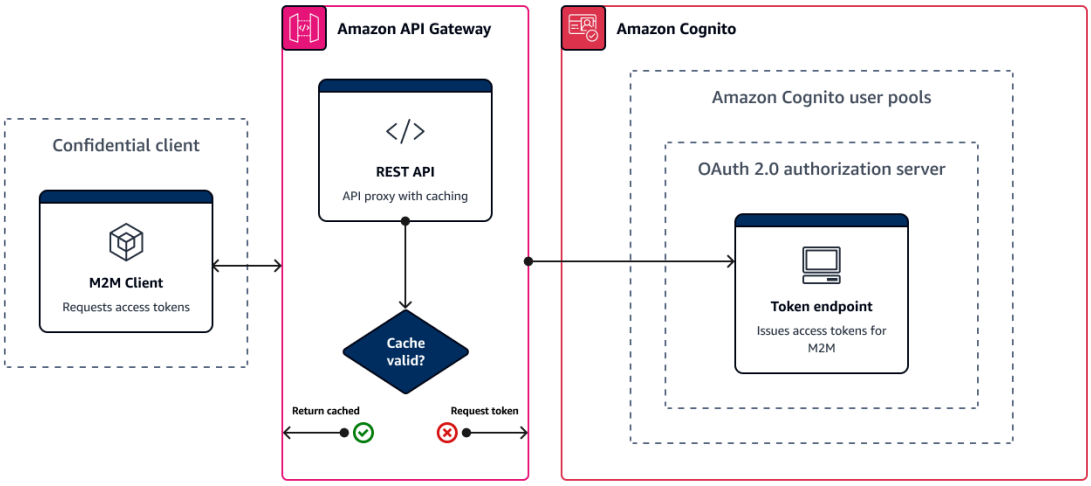
Puede configurar el grupo de usuarios para que los tokens caduquen en minutos, horas o días. Para garantizar el rendimiento y la disponibilidad de la aplicación, utilice los tokens de Amazon Cognito durante aproximadamente el 75 % de la vida útil del token y solo entonces recupere los nuevos. Una solución 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 volumen elevado de actividad de usuario o de máquina a máquina, es posible que encuentre 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 del inicio de sesión administrado. 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 tokens de acceso de máquina a máquina 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 Gateway-based solución de API ofrece una implementación de almacenamiento en caché de tokens de baja latencia y bajo contenido code/no de 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 cliente de OAuth 2.0, un tipo de concesión con frecuencia de gran volumen que produce tokens de acceso para autorizar sesiones de máquina a máquina y de microservicio. En el caso de que se produzca un aumento de tráfico que provoque que sus microservicios escalen horizontalmente, es posible que muchos sistemas utilicen las mismas credenciales de cliente con un volumen que supere el límite de la AWS tasa de solicitudes de su grupo de usuarios o del 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 `scope` del cuerpo de la solicitud y en 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 puede implementar el almacenamiento en caché de los tokens con ElastiCache (Redis OSS). 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 costo 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.

   > **Note**
   >
   > Este método de validación de solicitudes se utiliza con la `client_secret_basic` autorización en las solicitudes de token, donde el secreto del cliente está codificado en el encabezado de la solicitud `Authorization`. Para validar el cuerpo de la solicitud JSON en la autorización `client_secret_post`, cree en su lugar un modelo de datos que requiera la presencia de client_secret. En este modelo, el validador de solicitudes debe validar el cuerpo, los parámetros de la cadena de consulta y los encabezados.

   b. Configure el método Validador de solicitudes para validar los encabezados y parámetros de la cadena de consulta. Para obtener más información sobre la validación de solicitudes, consulte Validación de solicitud en la Guía para desarrolladores de Amazon API Gateway.

   c. Establezca su parámetro `scope` y el encabezado `Authorization` como clave de almacenamiento en caché.

      i. Añada una cadena de consulta a los parámetros de la cadena de consulta URL. Introduzca el nombre de una cadena de consulta de `scope` y seleccione Obligatorio y Almacenamiento en caché.

      ii. Agregue un encabezado en los Encabezados de solicitud HTTP. Introduzca el nombre de un encabezado de solicitud de `Authorization` y seleccione Obligatorio y Almacenamiento en caché.

4. En Stages (Etapas), configure el almacenamiento en caché.

   a. Elija la etapa que desee modificar y seleccione Editar en Detalles de la etapa.

   b. En Configuración adicional, para Configuración de caché, active Aprovisionar caché de API.

   c. Elija una Cache capacity (Capacidad de caché). Una mayor capacidad de caché mejora el rendimiento, pero conlleva un costo adicional.

   d. Desmarque la casilla de verificación Require authorization (Requerir autorización). Seleccione Continuar.

   e. API Gateway solo aplica políticas de caché a los métodos GET desde el nivel de etapa. Debe aplicar una anulación de la política de caché a su método POST.

Amplíe la etapa que configuró y seleccione el método `POST`. Para crear la configuración de caché para el método, elija Crear anulación.

f. Active la opción Habilitar caché de métodos.

g. Introduzca un Tiempo de vida (TTL) de caché de 3600 segundos. Seleccione Save.

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.

Ámbitos, M2M y 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 obtener más información, consulte Inicio de sesión administrado de grupos de usuarios. 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 se indican en la notificación `scope` del token de acceso. Para obtener más información sobre las solicitudes de tokens de acceso a Amazon Cognito, consulte Descripción del token de acceso.

Con Amazon Cognito, los ámbitos de los tokens de acceso pueden autorizar el acceso a API externas o a atributos del usuario. Puede emitir tokens de acceso a usuarios locales, usuarios federados o identidades de máquinas.

Temas

- Autorización de API
- Machine-to-machine autorización (M2M)
- Acerca de los ámbitos
- Acerca de los servidores de recursos

- Vinculación de recursos

Autorización de API

A continuación, indicamos varias formas de autorizar solicitudes a API con los tokens de Amazon Cognito:

Token de acceso

Cuando añada un autorizador de Amazon Cognito a una configuración de solicitud de método de la API de REST, añada ámbitos de autorización a la configuración del autorizador. Con esta configuración, la API acepta los tokens de acceso del encabezado `Authorization` y busca en ellos los ámbitos aceptados.

Token de ID

Cuando pasa un token de ID válido a un autorizador de Amazon Cognito en su API REST, API Gateway acepta la solicitud y pasa el contenido del token de ID al back-end de la API.

Amazon Verified Permissions

En Verified Permissions, tiene la opción de crear un almacén de API-linked políticas. Verified Permissions crea y asigna un autorizador de Lambda que procesa los tokens de ID o de acceso del encabezado `Authorization` de la solicitud. Este autorizador de Lambda pasa el token a su almacén de políticas, donde Verified Permissions lo compara con las políticas y devuelve al autorizador una decisión de permitir o denegar.

Más recursos

- Control y administración del acceso a una API de REST en API Gateway
- Autorización con Amazon Verified Permissions

Machine-to-machine autorización (M2M)

Amazon Cognito admite aplicaciones que acceden a los datos de la API con identidades de máquinas. Las identidades de máquinas en los grupos de usuarios son clientes confidenciales que se ejecutan en servidores de aplicaciones y se conectan a API remotas. En su funcionamiento no interviene el usuario; se trata de tareas programadas, flujos de datos o actualizaciones de activos. Cuando estos clientes autorizan sus solicitudes con un token de acceso, realizan una autorización de

máquina a máquina o M2M. En la autorización de M2M, se reemplazan las credenciales de usuario por un secreto compartido en el control de acceso.

Una aplicación que acceda a una API con una autorización de M2M debe tener un ID y un secreto de cliente. En su grupo de usuarios, cree un cliente de aplicaciones que tenga un secreto de cliente. De este modo, la identidad de su máquina puede obtener un token de acceso M2M de dos maneras: puede solicitar la Punto de conexión de tokenconcesión de credenciales de cliente o puede llamar a la operación de la GetClientTokenAPI. Ambos enfoques emiten un token de acceso que autoriza únicamente los ámbitos personalizados de los servidores de recursos. Se diferencian en su configuración: el terminal del token requiere un dominio de grupo de usuarios y se adapta a las aplicaciones que utilizan una biblioteca OIDC, mientras que no `GetClientToken` requiere ningún dominio y funciona mediante el AWS SDK o la API. AWS CLI Para obtener más información acerca de la configuración de clientes de aplicación, consulte Application-specific configuración con clientes de aplicaciones.

El token de acceso M2M es una declaración verificable de las operaciones que quieres permitir que la identidad de tu máquina solicite a una API. Para obtener más información sobre cómo los tokens de acceso autorizan las solicitudes de API, siga leyendo. Para ver un ejemplo de aplicación, consulte Amazon Cognito and API Gateway based machine to machine authorization using AWS CDK.

La autorización de M2M tiene un modelo de facturación que difiere de la forma en que se facturan los usuarios activos mensuales (MAU). Si bien la autenticación de los usuarios conlleva un coste por usuario activo, la facturación M2M refleja los clientes de aplicaciones M2M activos, ya sea que utilicen la concesión de credenciales de cliente o el funcionamiento de la `GetClientToken` API, y el volumen total de solicitudes de tokens. Para obtener más información, consulte Precios de Amazon Cognito. Para controlar los costos que generan las autorizaciones de M2M, optimice la duración de los tokens de acceso y el número de solicitudes de token que realizan las aplicaciones. Consulte Administración de la caducidad y el almacenamiento en caché de los tokens del grupo de usuarios para ver una forma de utilizar el almacenamiento en caché de API Gateway para reducir las solicitudes de nuevos tokens en las autorizaciones de M2M.

Para obtener información sobre cómo optimizar las operaciones de Amazon Cognito que añaden costes a su AWS factura, consulte. Administración de costos

Metadatos del cliente para credenciales de cliente máquina a máquina (M2M)

Puede pasar los metadatos del cliente en las solicitudes M2M. Los metadatos del cliente son información adicional de un entorno de usuario o aplicación que puede contribuir a los resultados de una Desencadenador de Lambda anterior a la generación del token. En las operaciones de

autenticación con un usuario principal, puede pasar los metadatos del cliente al activador previo a la generación del token en el cuerpo de AdminRespondToAuthChallengelas solicitudes a la RespondToAuthChallengeAPI. Dado que las aplicaciones dirigen el flujo de generación de tokens de acceso para M2M con solicitudes directas al Punto de conexión de token, tienen un modelo diferente. En el cuerpo POST de las solicitudes de token de credenciales de cliente, pasa un `aws_client_metadata` parámetro con el objeto de metadatos del cliente URL-encoded (`x-www-form-urlencoded`) a la cadena. Para obtener una solicitud de ejemplo, consulte Credenciales de cliente con autorización básica. A continuación, puede ver un ejemplo de parámetro que transfiere los pares clave-valor `{"environment": "dev", "language": "en-US"}`.

```
aws_client_metadata=%7B%22environment%22%3A%20%22dev%22,%20%22language%22%3A%20%22en-US
%22%7D
```

Obtener los tokens de acceso M2M con GetClientToken

La operación de la GetClientTokenAPI emite un token de acceso M2M para un cliente de aplicación confidencial a través del AWS SDK, o API AWS CLI, sin un dominio de grupo de usuarios. Proporciona la misma funcionalidad que las credenciales del cliente conceden desde el punto final del token; ambas autorizan una aplicación y no a un usuario.

> **Note**
>
> Amazon Cognito no utiliza políticas basadas en la identidad AWS Identity and Access Management (IAM) para autorizar. `GetClientToken` No puede usar las credenciales de IAM para autorizar la solicitud; la autorización proviene del secreto de cliente del cliente de la aplicación.

Para usarlo`GetClientToken`, configura un cliente de aplicación que tenga un secreto de cliente y solo el flujo de `ALLOW_CLIENT_TOKEN_AUTH` autenticación. Este flujo se excluye mutuamente de los flujos de autenticación de usuarios. Un cliente de aplicación puede tener hasta dos secretos activos. Asocie los ámbitos personalizados del servidor de recursos al cliente de la aplicación. En el siguiente AWS CLI ejemplo, se crea un cliente de aplicaciones compatible.

```
aws cognito-idp create-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-name my-m2m-client \
    --generate-secret \
    --explicit-auth-flows ALLOW_CLIENT_TOKEN_AUTH \
```

```
    --allowed-o-auth-scopes "solar-system-data/asteroids.add"
```

Para obtener un token de acceso, llama `GetClientToken` con el ID del cliente de la aplicación, un secreto de cliente activo y los ámbitos personalizados que quieras incluir en el token. `GetClientToken`no toma ningún `UserPoolId` parámetro; Amazon Cognito identifica el grupo de usuarios a partir del ID de cliente. Si no lo especifica`Scopes`, Amazon Cognito autoriza los ámbitos configurados para el cliente de la aplicación.

```
aws cognito-idp get-client-token \
    --client-id 1example23456789 \
    --secret exampleClientSecret123EXAMPLE \
    --scopes "solar-system-data/asteroids.add"
```

La respuesta contiene el token de acceso y sus metadatos en un objeto. `ClientAuthenticationResult`

```
{
    "ClientAuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer"
    }
}
```

Presente el token de acceso a su servidor de recursos, por ejemplo, en el `Authorization` encabezado de una solicitud a una API REST de Amazon API Gateway. El servidor de recursos verifica la firma y el vencimiento del token y, a continuación, autoriza la solicitud en función de los ámbitos del token.

Para pasar datos a una entidad Desencadenador de Lambda anterior a la generación del token que personalice el token, incluye un `ClientMetadata` mapa en la solicitud con el parámetro. `--client-metadata` El activador previo a la generación del token debe estar configurado para la versión 3 o posterior del evento desencadenante para recibir los metadatos del cliente en el flujo M2M. Amazon Cognito no almacena, valida ni cifra estos datos y solo los pone a disposición de los activadores de Lambda, por lo que no incluya información confidencial.

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 para los grupos de usuarios, los ámbitos personalizados y los ámbitos de OpenID Connect (OIDC).

Ámbito de API reservado para los grupos de usuarios

El ámbito `aws.cognito.signin.user.admin` autoriza al usuario actual las operaciones de autoservicio en la API del grupo de usuarios de Amazon Cognito. Autoriza al portador de un token de acceso a consultar y actualizar toda la información sobre el portador con, por ejemplo, las operaciones de la API y la API. GetUserUpdateUserAttributes 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 El punto de conexión 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 de OpenID Connect (OIDC)

Cuando autentique usuarios con el servidor de autorización del grupo de usuarios, incluso con el inicio de sesión administrado, debe 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 de OIDC autorizan a su aplicación a leer información del usuario en el El punto de conexión 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 aplicación a emitir tokens de acceso con los siguientes ámbitos de OIDC.

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 de OIDC como `profile`, `email` y `phone`, el contenido del token de ID y la respuesta 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 El punto de conexión 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.

correo electrónico

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.

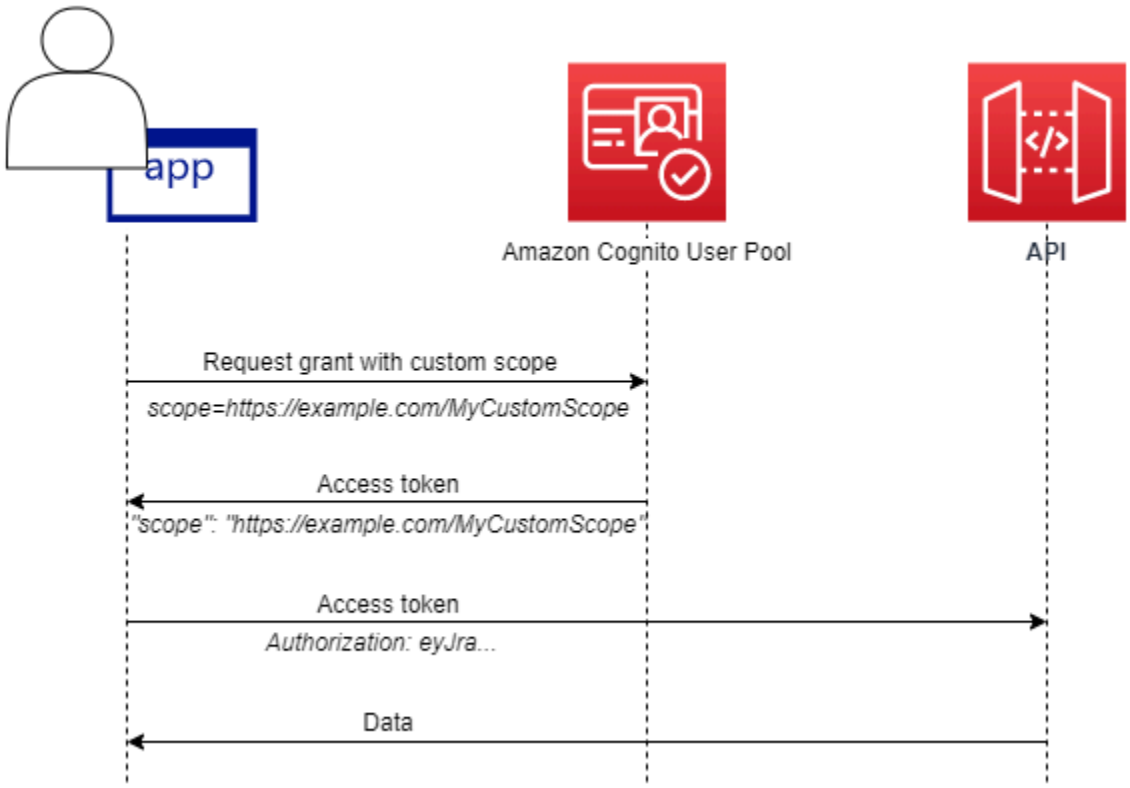
Usted define los ámbitos personalizados, que amplían las capacidades de autorización de un grupo de usuarios para incluir fines no relacionados con la consulta y modificación de usuarios y sus atributos. Por ejemplo, si tiene un servidor de recursos para fotos, podría definir dos ámbitos: `photos.read` para el acceso de lectura a las fotos y `photos.write` para write/delete el acceso. 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 tokens web JSON. 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.

Descripción general de

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. En el caso de M2M, también puedes solicitar ámbitos personalizados con la operación de la GetClientTokenAPI. 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, tu aplicación puede solicitar el Punto de conexión de token canje de un código de autorización o la concesión de credenciales de cliente. En el inicio de sesión administrado, también puede solicitar ámbitos personalizados en un token de acceso a partir de una concesión implícita. En el caso de la autorización M2M, tu aplicación también puede solicitar ámbitos personalizados con la operación de la API. GetClientToken

> **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 el inicio de sesión administrado. 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 solicites el token de acceso que te autoriza a registrar tu descubrimiento, formatea el parámetro de solicitud `scope` HTTPS 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 Consola de administración de AWS CLI para definir los servidores de recursos y los ámbitos de su grupo de usuarios.

Definir un servidor de recursos para su grupo de usuarios (Consola de administración de AWS)

Puede usar el Consola de administración de AWS 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 User Pools (Grupos de usuarios), y elija el grupo de usuarios que desea editar.
3. Seleccione el menú Dominio en Creación de marca y localice los Servidores 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 el menú Dominio bajo Servidores de recursos, en la columna Ámbitos personalizados. Los ámbitos personalizados se pueden habilitar para clientes de aplicación desde el menú Clientes de aplicación, en Aplicaciones. Seleccione un cliente de aplicación, busque Páginas de inicio de sesión y elija Editar. AñadirÁmbitos personalizadosy eligeGuarde los cambios.

Definir un servidor de recursos para su grupo de usuarios (AWS CLI and 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

Eliminación de 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

Vinculación de recursos

Con la vinculación de recursos, también conocida como indicadores de recursos, puede solicitar API-specific concesiones al servidor de autorización de su grupo de usuarios. La vinculación de recursos es una extensión de OAuth 2.0 definida en RFC 8707, y permite a los clientes especificar explícitamente a qué servidor de recursos desean acceder durante las solicitudes de autorización.

Con la vinculación de recursos, las configuraciones de la API pueden denegar el acceso a los tokens que no estén diseñados específicamente para ellos.

> **Note**
>
> Solo puede vincular los tokens de acceso a los recursos de los usuarios. No puede solicitar una vinculación de recursos con las credenciales de los clientes que concede M2M.

Al utilizar la vinculación de recursos con los grupos de usuarios de Amazon Cognito, los clientes pueden incluir un parámetro `resource` en sus solicitudes de autenticación al servidor de autorización del grupo de usuarios. Su grupo de usuarios valida que el valor del recurso solicitado es una URL y sigue las mismas reglas de esquema que las URL de devolución de llamadas de los clientes de aplicación: `https://`, `http://` solo con `localhost` o un esquema personalizado, como `myapp://`. Amazon Cognito establece el URI solicitado como la audiencia en la reclamación `aud` del token de acceso. Si el recurso solicitado es un servidor de recursos de un grupo de usuarios, el identificador del servidor de recursos debe tener un formato de URL. Puede solicitar un recurso por cada solicitud de autenticación.

Esta característica está destinada a la autenticación del inicio de sesión administrado con el servidor de autorización OAuth 2.0 del grupo de usuarios. Puede solicitar la vinculación de recursos mediante concesiones implícitas y de códigos de autorización desde. Autorizar punto de conexión Token-refresh las subvenciones de la solicitud original se Punto de conexión de token transfieren de la `aud` solicitud original. Actualmente, no está disponible en los modelos de autenticación del SDK.

Implementación de la vinculación de recursos con grupos de usuarios de Amazon Cognito

1. Configure uno o más servidores de recursos en su grupo de usuarios con identificadores únicos.
2. En su solicitud de autorización a `/oauth2/authorize`, solicite un código de autorización o una concesión implícita e incluya el parámetro `resource`. El valor de `resource` debe ser un identificador del servidor de URL-formatted recursos o una URL. Por ejemplo, `&resource=https://solar-system-data-api.example.com`.
3. El servidor de autorización valida la solicitud de recurso, completa la autenticación y establece la reclamación `aud` del token de acceso en la URL del recurso solicitado.
4. Para validar que los tokens se emitieron específicamente para esta, el recurso que consume el token de acceso del usuario comprueba la reclamación `aud`.

Administrar usuarios

Una vez que los usuarios puedan iniciar sesión, usted administrará sus cuentas durante la vigencia de la aplicación. En esta sección se describe la administración diaria de un directorio de usuarios: crear e importar usuarios, organizarlos en grupos y conectar un grupo de usuarios a un grupo de identidades para que los usuarios también puedan obtener AWS credenciales.

Temas

- Administración de usuarios en el grupo de usuarios
- Actualización de la configuración del grupo de usuarios y del cliente de aplicación

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.

Dispone de varias opciones de configuración y tareas administrativas para administrar los usuarios de un grupo de usuarios de Amazon Cognito. Los grupos de usuarios pueden escalarse hasta alcanzar millones de usuarios. Un directorio de usuarios de esta magnitud requiere herramientas administrativas que también sean escalables y puedan repetirse. Puede que quiera crear numerosos perfiles de usuario, administrar los usuarios inactivos, elaborar informes de gobernanza y conformidad o configurar herramientas de autoservicio en las que los usuarios realicen la mayor parte del trabajo. Después de crear un grupo de usuarios, puede controlar el modo en que los usuarios se registran y confirman sus cuentas, como, por ejemplo, exigir una verificación por correo electrónico o número de teléfono. Los administradores también pueden crear directamente cuentas de usuario y personalizar los mensajes de bienvenida y los requisitos de contraseña.

Los grupos de usuarios tienen, a su vez, grupos de usuarios, en los que puede administrar el acceso a los recursos en función de la pertenencia a un determinado grupo de usuarios. Puede asignar roles de IAM a estos grupos para administrar el acceso a los Servicios de AWS con grupos de identidades. La pertenencia a un determinado grupo de usuarios se incluye en el token de ID y en el de acceso. Con esta información, puede tomar decisiones de control de acceso en tiempo de ejecución en la aplicación o con un motor de políticas como Amazon Verified Permissions.

Los grupos de usuarios suelen tener numerosos usuarios. Con frecuencia tendrá que buscar y actualizar cuentas de usuario. La API y la consola de Amazon Cognito permiten realizar consultas

a los usuarios en función de atributos estándar, como el nombre de usuario, el correo electrónico o el número de teléfono. Los administradores también pueden restablecer contraseñas, deshabilitar cuentas y ver el historial de eventos de los usuarios.

Para migrar los datos de los usuarios ya existentes, Amazon Cognito dispone de opciones para importar usuarios desde un archivo CSV y utilizar un desencadenador de Lambda para migrar automáticamente a los usuarios cuando estos inicien sesión por primera vez. Estas opciones permiten que los usuarios de otros directorios de usuarios puedan pasar a su grupo de usuarios.

Puede utilizar las características de administración de usuarios de los grupos de usuarios para controlar de forma pormenorizada el ciclo de vida de los usuarios y la experiencia de autenticación. La combinación del registro de autoservicio, las cuentas creadas por el administrador, los grupos y las herramientas de migración convierte a los grupos de usuarios de Amazon Cognito en un directorio de usuarios flexible.

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
- Contraseñas, recuperación de contraseñas y políticas de contraseñas
- Importación de usuarios a un grupo de usuarios
- Uso de atributos de usuario

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 aplicación, con o sin inicio de sesión administrado, crea un perfil de usuario para cualquier persona en 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.

Consola de administración de AWS

El Sign-upmenú de tu grupo de usuarios contiene algunos de los ajustes para el registro y la creación administrativa de los usuarios de tu grupo de usuarios.

Para configurar la experiencia de registro

1. En Cognito-assisted Verificación y confirmación, elija si quiere 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 al crear un 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. Al Self-service registrarse, seleccione Habilitar el registro automático si desea que los usuarios puedan generar una cuenta nueva con la API no autenticada. `SignUp` Si inhabilita el registro automático, solo podrá crear nuevos usuarios como administrador, en la consola de Amazon Cognito o AdminCreateUsercon solicitudes de API. En un grupo de usuarios en el que el registro automático está inactivo, las solicitudes de SignUpAPI se devuelven `NotAuthorizedException` y el inicio de sesión gestionado no muestra el enlace de 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 los ajustes del menú Métodos de autenticación, 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 el menú Métodos de autenticación, 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 aplicación con secretos de cliente en el menú Clientes de aplicación 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 API CreateUserPoolo UpdateUserPooluna solicitud de API.

El AdminCreateUserConfigelemento 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 CreateUserPoolsolicitud de UpdateUserPoolAPI 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.

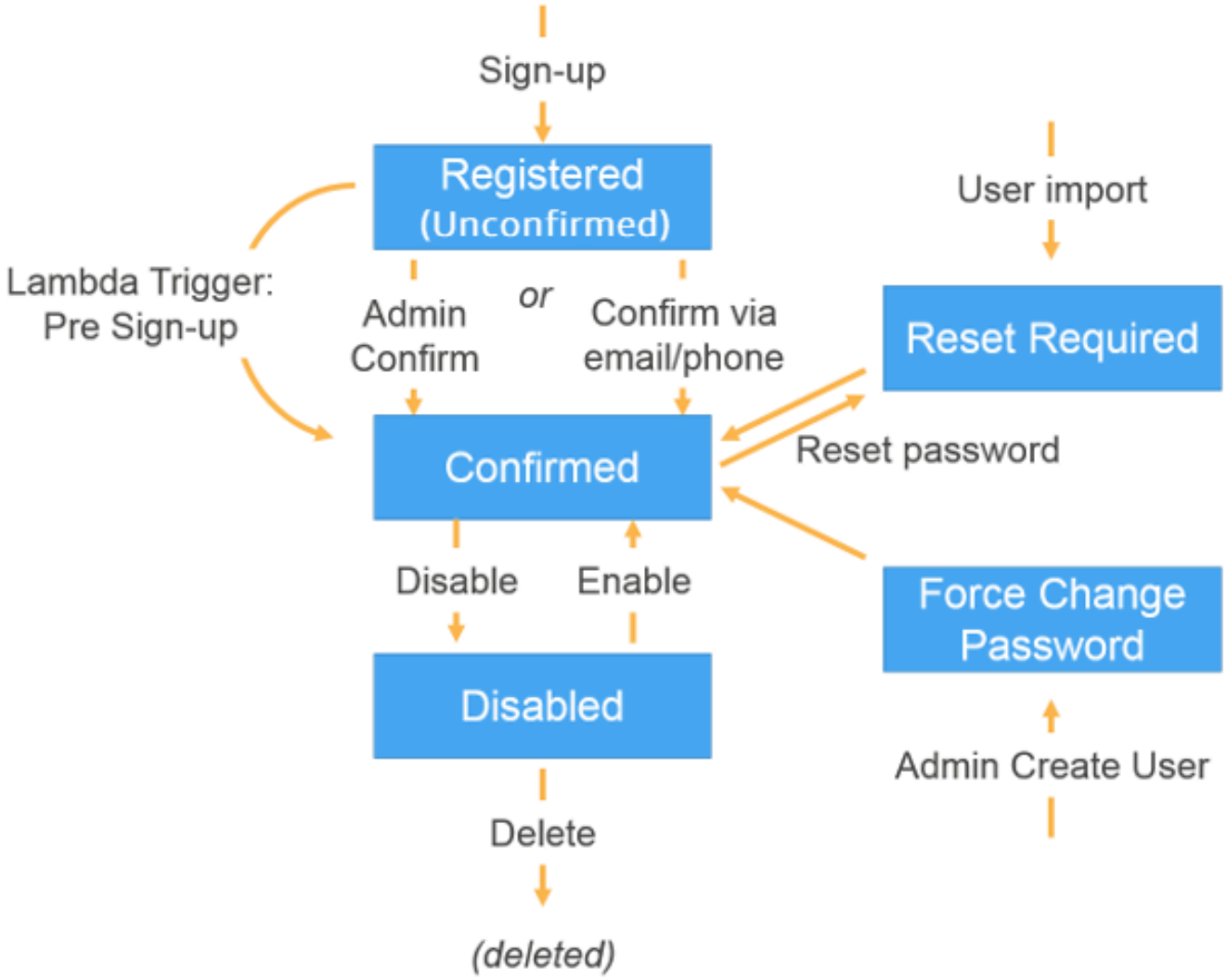
Contraseñas en el registro

Amazon Cognito exige contraseñas a todos los usuarios cuando se registran, excepto en las siguientes condiciones. Si se cumplen todas estas condiciones, puede omitir las contraseñas en las operaciones de registro.

1. El inicio de sesión sin contraseña está activo en su grupo de usuarios y en el cliente de aplicación.
2. La aplicación está diseñada a medida con módulos de autenticación en un AWS SDK. El inicio de sesión administrado y la interfaz de usuario alojada siempre requieren contraseñas.
3. Los usuarios proporcionan valores de atributo para los métodos de inicio de sesión sin contraseña, (contraseñas de un solo uso (OTP) por correo electrónico o mensaje SMS, que usted permita. Por ejemplo, si permite iniciar sesión con OTP por correo electrónico y teléfono, los usuarios pueden proporcionar un número de teléfono o una dirección de correo electrónico, pero si solo permite el inicio de sesión con correo electrónico, deberán proporcionar una dirección de correo electrónico.
4. Su grupo de usuarios verifica automáticamente los atributos que los usuarios pueden usar al iniciar sesión sin contraseña.
5. Para SignUpuna solicitud determinada, el usuario no proporciona un valor para el parámetro Password.

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 desencadenador de Lambda Antes del registro 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.

Deshabilitado

Para poder eliminar una cuenta de usuario, debe deshabilitar el acceso de inicio de sesión para ese usuario.

Más recursos

- Detección y corrección de cuentas de usuario inactivas con Amazon Cognito

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 transcurrido 24 horas y el código o enlace de tu usuario ya no es válido, llama a la operación de la 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.

Otras acciones que confirman y verifican los atributos del usuario

La siguiente actividad del usuario verifica los atributos del usuario. No es necesario configurar estos atributos para que se verifiquen automáticamente: las acciones enumeradas los marcan como verificados en todos los casos.

Dirección de correo electrónico

1. Completar correctamente la autenticación sin contraseña con una contraseña de un solo uso (OTP) por correo electrónico.

2. Completar correctamente la autenticación multifactor (MFA) con una OTP de correo electrónico.

Número de teléfono

1. Completar correctamente la autenticación sin contraseña con una OTP por SMS.
2. Completar correctamente la MFA con una OTP por SMS.

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 Consola de administración de AWS 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 Creación de una nueva aplicación en la consola de Amazon Cognito.

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. Seleccione el Sign-upmenú y localice la verificación de atributos y la confirmación de la cuenta de usuario. Elija Edit (Edición de).
4. En Cognito-assisted Verificación y confirmación, elija si permitirá 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 puede deshabilitar la Cognito-assisted verificación y la confirmación y utilizar acciones de API autenticadas o activadores Lambda para verificar los atributos y confirmar a los 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 de Lambda Antes del 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 son el atributo que se verifica cuando se confirman las cuentas; debes tomar medidas adicionales para verificar las direcciones de correo electrónico de los usuarios y comunicarte con ellas.

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 son el atributo que se verifica cuando se confirman las cuentas; debe tomar medidas adicionales para verificar los números de teléfono de los usuarios y comunicarse con ellos.

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 se registra en tu aplicación introduciendo un nombre de usuario, un número de teléfono, una dirección de and/or 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 la IAM-authenticated API. 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 `Permanent` establecido en. `True`

1. Un usuario se registra en tu aplicación introduciendo un nombre de usuario, un número de teléfono, una dirección de and/or 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 Consola de administración de AWS o firmar tu solicitud de API con AWS credenciales para confirmar la cuenta.

    a. Para confirmar un usuario en la consola de Amazon Cognito, vaya al menú 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-up en. 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 la concatenación, `HMAC_SHA256` representa una función que produce un valor de HMAC mediante HMACSHA256 y representa una función que produce una versión de la salida del hash. `Base64` Base-64-encoded

```
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 Antes del registro 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 Antes del registro, 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

En los grupos de usuarios que se configuran con varios nombres de inicio de sesión, los usuarios pueden introducir un número de teléfono o una dirección de correo electrónico como nombre de usuario al iniciar sesión. Cuando actualizan su dirección de correo electrónico o número de teléfono en la aplicación, Amazon Cognito puede enviarles inmediatamente un mensaje con un código que verifica que son propietarios del nuevo valor de atributo. Para habilitar el envío automático de estos códigos de verificación, consulte Configuración de la verificación del correo electrónico o del teléfono.

Los usuarios que reciban un código de verificación deberán devolverlo a Amazon Cognito en una VerifyUserAttributesolicitud. Tras proporcionar el código, su atributo se marca como verificado. Normalmente, cuando los usuarios actualizan su dirección de correo electrónico o número de teléfono, es recomendable comprobar que son los propietarios del nuevo valor antes de poder usarlo para iniciar sesión y recibir mensajes. Los grupos de usuarios tienen una opción configurable que determina si los usuarios deben comprobar las actualizaciones de su dirección de correo electrónico o número de teléfono.

Esta opción es la propiedad `AttributesRequireVerificationBeforeUpdate` del grupo de usuarios. Configúralo en una UpdateUserPoolsolicitud CreateUserPoolo con la opción Mantener activo el valor del atributo original cuando haya una actualización pendiente en el Sign-upmenú de la consola de Amazon Cognito.

La forma en que su grupo de usuarios trate las actualizaciones de las direcciones de correo electrónico y los números de teléfono está relacionada con la configuración de nombres de usuario de su grupo de usuarios. Los nombres de usuario del grupo de usuarios pueden estar en una configuración de atributos de nombre de usuario en la que los nombres de inicio de sesión sean la dirección de correo electrónico, el número de teléfono o ambas cosas. También pueden estar en una configuración de atributos de alias en la que el atributo `username` sea un nombre de inicio de sesión junto con una dirección de correo electrónico, un número de teléfono o un nombre de usuario preferido como nombres de inicio de sesión alternativos. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

También 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 estén verificados, su aplicación debe informar al usuario de que debe verificar el atributo y proporcionar un botón o enlace para que los usuarios introduzcan su código de verificación.

En la siguiente tabla, se describe cómo determinan `AttributesRequireVerificationBeforeUpdate` y la configuración del alias el resultado cuando los usuarios cambian el valor de sus atributos de inicio de sesión.

Configuración de nombre de usuario	Comportamiento cuando los usuarios deben verificar nuevos atributos	Comportamiento cuando los usuarios no tienen la obligación de verificar nuevos atributos
Atributos de nombre de usuario	El atributo original permanece verificado, apto para iniciar sesión y mantiene su valor original. Cuando el usuario verifica un valor nuevo, Amazon Cognito actualiza el valor del atributo, lo marca como verificado y lo convierte en apto para iniciar sesión.	Amazon Cognito actualiza el atributo a un nuevo valor. El nuevo valor es apto para iniciar sesión. Cuando el usuario verifica un valor nuevo, Amazon Cognito lo marca como verificado.
Atributos de alias	El atributo original permanece verificado, apto para iniciar sesión y mantiene su valor original. Cuando el usuario verifica un valor nuevo, Amazon Cognito actualiza el valor del atributo, lo marca como verificado y lo convierte en apto para iniciar sesión.	Amazon Cognito actualiza el atributo a un nuevo valor. Ni el valor de atributo original ni el nuevo son aptos para iniciar sesión. Cuando el usuario verifica un valor nuevo, Amazon Cognito actualiza el valor del atributo, lo marca como verificado y lo convierte en apto para iniciar sesión.

Ejemplo 1

El usuario 1 inicia sesión en la aplicación con la dirección de correo electrónico `user1@example.com` y tiene el nombre de usuario `user1` (atributos de alias). Su grupo de usuarios está configurado para verificar las actualizaciones de los atributos de inicio de sesión y enviar automáticamente los mensajes de verificación. Solicitan actualizar su dirección de correo electrónico a `user1+foo@example.com`. Reciben un correo electrónico de verificación en `user1+foo@example.com` y solo pueden volver a iniciar sesión con esa dirección `user1@example.com`. Más tarde, introducen su código de verificación y solo pueden volver a iniciar sesión con la dirección de correo electrónico `user1+foo@example.com`.

Ejemplo 2

El usuario 2 inicia sesión en la aplicación con la dirección de correo electrónico `user2@example.com` y tiene un nombre de usuario (atributos de alias). Su grupo de usuarios está configurado para no verificar las actualizaciones de los atributos de inicio de sesión y para enviar automáticamente los mensajes de verificación. Solicitan actualizar su dirección de correo electrónico a `user2+bar@example.com`. Reciben un correo electrónico de verificación en `user2+bar@example.com` y no pueden volver a iniciar sesión. Más tarde, introducen su código de verificación y solo pueden volver a iniciar sesión con la dirección de correo electrónico `user2+bar@example.com`.

Ejemplo 3

El usuario 3 inicia sesión en la aplicación con la dirección de correo electrónico `user3@example.com` y no tiene un nombre de usuario (atributos de nombre de usuario). Su grupo de usuarios está configurado para no verificar las actualizaciones de los atributos de inicio de sesión y para enviar automáticamente los mensajes de verificación. Solicitan actualizar su dirección de correo electrónico a `user3+baz@example.com`. Reciben un correo electrónico de verificación en `user3+baz@example.com`, pero pueden iniciar sesión inmediatamente sin necesidad de realizar ninguna otra acción con el código de verificación.

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 Consola de administración de AWS 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 el menú Métodos de autenticación. 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 el menú Plantillas de mensajes 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, consulta ¿Qué son las cuotas de gasto a nivel de cuenta y de mensaje y cómo funcionan? en Amazon SNS FAQs.

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 icono Región de AWS para mensajes SMS.

Región de Amazon Cognito	Región alternativa de Amazon SNS heredada
Este de EE. UU. (Ohio)	Este de EE. UU. (Norte de Virginia)
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. En el menú Registro, seleccione Editar, en Verificación de atributos y confirmación de la 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. Seleccione Save changes (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 el menú Métodos de autenticación de la consola de Amazon Cognito, en SMS, elija Editar para configurar un rol.

Configuración de los mensajes de verificación, invitación, autenticación y MFA

Con Amazon Cognito, puede personalizar los mensajes de autenticación, verificación e invitación a usuarios por SMS y correo electrónico para mejorar la seguridad y la experiencia del usuario de su aplicación. Puede elegir entre verificaciones de código o de enlace con un solo clic para algunos mensajes. En este tema, se explica cómo personalizar la autenticación y las comunicaciones de verificación en la consola de Amazon Cognito.

En el menú Plantillas de mensajes, puede personalizar lo siguiente:

- Sus plantillas de correo electrónico y mensajes SMS para la autenticación multifactor (MFA) y mediante contraseña de un solo uso (OTP).
- Los mensajes de verificación de SMS y correo electrónico
- El tipo de verificación por correo electrónico: código o enlace

> **Note**
>
> Amazon Cognito envía enlaces con su plantilla basada en enlaces en los mensajes de verificación cuando los usuarios se registran o reenvían un código de confirmación. Los correos electrónicos de las operaciones de actualización de atributos y restablecimiento de contraseñas utilizan la plantilla de código.

- Los mensajes de invitación al usuario
- DIRECCIONES DE ORIGEN y de REPLY-TO correo electrónico para los correos electrónicos que pasan por su grupo de usuarios

> **Note**
>
> Las plantillas de mensajes de verificación por SMS y de correo electrónico solo aparecen si ha elegido exigir la verificación de número de teléfono y de correo electrónico. 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 de mensajes de correo electrónico y SMS para la MFA
- 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 marcadores de posición en los mensajes. Amazon Cognito sustituye dichos marcadores de posición con los valores que correspondan. Puede hacer referencia a los marcadores de posición de plantillas universales en las plantillas de mensajes de cualquier tipo, aunque estos valores no estén presentes en todos los tipos de mensajes.

Marcadores de posición de plantillas universales

Description (Descripción)	Token	Tipo de mensaje
Código de verificación	`{####}`	Mensajes de verificación, confirmación y MFA
Contraseña temporal	`{####}`	Forgot-password y mensajes de invitación
Nombre de usuario	`{username}`	Mensajes de invitación y de seguridad avanzada

Una de las respuestas automatizadas disponibles con protección contra amenazas consiste en notificar al usuario que Amazon Cognito ha detectado una actividad potencialmente maliciosa. 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. La protección contra amenazas de Amazon Cognito puede analizar estos detalles.
- Verificar si un enlace de un clic es válido.
- Utilice el identificador del evento, el token de comentarios y el nombre de usuario para crear su propio enlace con 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.

La protección contra amenazas añade los siguientes marcadores de posición que puede insertar en las plantillas de mensajes. Estos marcadores de posición se aplican a los mensajes de autenticación flexible, notificaciones que Amazon Cognito envía a los usuarios cuyas sesiones se han evaluado para determinar su nivel de riesgo. Para configurar las plantillas de mensajes con estas variables, actualice la Full-functionconfiguración de la protección contra amenazas en la consola de Amazon Cognito o envíe las plantillas en una SetRiskConfigurationsolicitud.

Marcadores de posición de las plantillas de seguridad avanzadas

Description (Descripción)	Token
Dirección IP	`{ip-address}`
Ciudad	`{city}`
País	`{country}`
Log-in hora	`{login-time}`
Nombre del dispositivo	`{device-name}`
One-click el enlace es válido	`{one-click-link-valid}`
One-click el enlace no es válido	`{one-click-link-invalid}`
ID de evento	`{event-id}`
Token de comentarios	`{feedback-token}`

Personalización de mensajes de correo electrónico y SMS para la MFA

A fin de personalizar los mensajes SMS y de correo electrónico para la autenticación multifactor (MFA), edite el mensaje de MFA en el menú Plantillas de mensajes de la consola de grupos de usuarios de Amazon Cognito.

> **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 establece una longitud máxima de 140 UTF-8 caracteres para los mensajes SMS, incluido el código de autenticación.

Personalización de los mensajes de verificación por SMS

A fin de personalizar el mensaje SMS para la verificación del número de teléfono, edite la plantilla Mensaje de verificación en el menú Plantillas de mensajes de su grupo de usuarios.

> **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, incluido el código de verificación, es de 140 UTF-8 caracteres.

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 Mensaje de verificación en el menú Plantillas de mensajes del grupo de usuarios. Puede elegir un Tipo de verificación de código o enlace al editar la plantilla 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, incluido el código de verificación (si lo hay), es de 20 000 UTF-8 caracteres. 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 Mensajes de invitación en la pestaña Plantillas de mensajes.

> **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 del usuario.

La longitud máxima de un mensaje SMS, incluido el código de verificación, es de 140 UTF-8 caracteres. La longitud máxima de un mensaje de correo electrónico, incluido el código de verificación, es de 20 000 UTF-8 caracteres. 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. Puedes elegir especificar direcciones de origen y REPLY-TO correo electrónico personalizadas en lugar de no-reply@verificationemail.com.

Para personalizar las direcciones de origen y REPLY-TO correo electrónico

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 el menú Métodos de autenticación. 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 usar una dirección de correo electrónico de un dominio verificado, configura los ajustes de correo electrónico en la API AWS Command Line Interface o en la AWS API. 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 REPLY-TO correo electrónico, introduce una dirección de correo electrónico válida en el campo de dirección de REPLY-TO correo electrónico.

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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "stmnt1234567891234",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "email.cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "arn:aws:ses:us-
east-1:111122223333:identity/support@example.com",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        }
    ]
}
```

La consola Amazon Cognito le añade una política similar cuando selecciona una identidad de Amazon SES en 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

Los grupos de usuarios no son solo un directorio de usuarios de administración de acceso e identidad de los clientes (CIAM) donde cualquier usuario de Internet puede registrarse para obtener un perfil de usuario en su aplicación. Puede deshabilitar el registro de autoservicio. Es posible que ya conozca a los clientes y solo desee admitir a aquellos que hayan sido autorizados previamente. Puede colocar barreras de protección de autenticación manual en torno a su aplicación con un proveedor de identidades SAML 2.0 u OIDC privado, importando usuarios, filtrándolos al registrarse o creando usuarios mediante operaciones administrativas de API. El flujo de trabajo de creación administrativa de los usuarios puede ejecutarse mediante programación, aprovisionando a los usuarios después de que se hayan registrado en otro sistema, o puede llevarse a cabo caso por caso o mediante pruebas en la consola de Amazon Cognito.

Al crear usuarios como administrador, Amazon Cognito establece una contraseña temporal para dichos usuarios y les envía un mensaje de bienvenida o de invitación. Los usuarios pueden seguir el enlace del mensaje de invitación e iniciar sesión por primera vez, establecer una contraseña y confirmar su cuenta. En la siguiente página se describe cómo crear usuarios nuevos y configurar el mensaje de bienvenida. Para obtener más información sobre la creación de usuarios con la API de grupos de usuarios y un AWS SDK o CDK, consulte AdminCreateUser.

> **Los usuarios con contraseñas temporales comienzan con el estado FORCE_CHANGE_PASSWORD**
>
> Al crear un usuario con una contraseña temporal, Amazon Cognito coloca al usuario en el `FORCE_CHANGE_PASSWORD` estado. Un usuario con este estado no puede iniciar sesión hasta que establezca una contraseña permanente en respuesta a un `NEW_PASSWORD_REQUIRED` desafío.

Para mover un usuario creado por el administrador al `CONFIRMED` estado sin iniciar sesión por primera vez, llama a la operación de la AdminSetUserPasswordAPI con el valor establecido en. `Permanent True` Esta solicitud establece una contraseña permanente y confirma al usuario.

Después de crear el grupo de usuarios, puede crear usuarios mediante la API Consola de administración de AWS Amazon Cognito 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 siguientes son algunos ejemplos de cómo los administradores pueden administrar los usuarios en los grupos de usuarios.

- Crear un perfil de usuario nuevo en la consola de Amazon Cognito o con la operación de la API `AdminCreateUser`.
- Poner a disposición de su grupo de usuarios y del cliente de aplicación los flujos de autenticación con nombre de usuario y contraseña, sin contraseña, con clave de acceso y personalizados.
- Establezca los valores de los atributos de usuario.
- Cree atributos personalizados.
- Establecer 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.
- Especificar una contraseña temporal, crear un usuario sin contraseña o permitir que Amazon Cognito genere una contraseña automáticamente.
- Crear nuevos usuarios y confirmar automáticamente sus cuentas, verificar sus direcciones de correo electrónico o sus números de teléfono.
- Especifique los mensajes de invitación por SMS y correo electrónico personalizados para los nuevos usuarios mediante los Consola de administración de AWS activadores Lambda, como el mensaje personalizado, el remitente de SMS personalizado y el remitente de correo electrónico personalizado.
- 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`.
- Suprimir el envío del mensaje de invitación cuando se crea el usuario.
- Especificar un límite de tiempo de caducidad de hasta 90 días para las cuentas de usuario nuevas.
- Permitir a los usuarios inscribirse o requerir que solo el administrador añada a los usuarios nuevos.

Los administradores también pueden iniciar sesión con AWS las credenciales de los usuarios en una aplicación del lado del servidor. Para obtener más información, consulte Modelos de autorización para la autenticación de API y SDK.

Flujos de autenticación de usuarios y creación de usuarios

La creación administrativa de usuarios tiene opciones que varían según la configuración del grupo de usuarios. Los flujos de autenticación, o los métodos disponibles para los usuarios para el inicio de sesión y la MFA, pueden cambiar la forma en que se crean los usuarios y los mensajes que se les envían. A continuación, se muestran algunos flujos de autenticación que están disponibles en los grupos de usuarios.

- Nombre de usuario y contraseña
- Clave de acceso
- Sign-in con un tercero IdPs
- Contraseñas de un solo uso (OTP) por correo o SMS sin contraseña establecida
- Multi-factor autenticación con correo electrónico, SMS y OTP de aplicaciones de autenticación
- Autenticación personalizada con activadores de Lambda

Para obtener más información sobre cómo configurar estos factores de inicio de sesión, consulte Autenticación con grupos de usuarios de Amazon Cognito.

Creación de usuarios sin contraseñas

Si ha activado el inicio de sesión sin contraseña en su grupo de usuarios, puede crear usuarios sin contraseñas. Para crear un usuario sin contraseña, debe proporcionar valores de atributo para un factor de inicio de sesión sin contraseña disponible. Por ejemplo, si el inicio de sesión con OTP por correo sin contraseña establecida está disponible en su grupo de usuarios, puede crear un usuario sin contraseña y con un atributo de dirección de correo electrónico. Si los únicos flujos de autenticación disponibles para los nuevos usuarios requieren una contraseña, por ejemplo, una clave de acceso o un nombre de usuario-contraseña, debe crear o generar una contraseña temporal para cada usuario nuevo.

Cómo crear un usuario nuevo sin contraseña

- Seleccione No establecer una contraseña en la consola de Amazon Cognito
- Omita o deje en blanco el parámetro `TemporaryPassword` de su solicitud de API `AdminCreateUser`

Los usuarios sin contraseña se confirman automáticamente

Normalmente, los nuevos usuarios obtienen una contraseña temporal y pasan a un estado `FORCE_CHANGE_PASSWORD` al crearlos. Cuando se crean usuarios sin contraseñas, pasan inmediatamente a un estado `CONFIRMED`. No puede reenviar los códigos de confirmación a estos usuarios en el estado `CONFIRMED`.

Los mensajes de invitación cambian para los usuarios sin contraseñas.

De forma predeterminada, Amazon Cognito envía un mensaje de invitación a los nuevos usuarios que dice `Your username is {userName} and your password is {####}.` Cuando crea usuarios sin contraseña, el mensaje dice `Your username is {userName}.` Personalice el mensaje de invitación para indicar si va a establecer contraseñas para los usuarios. Omita la variable de `{####}` contraseña en los modelos de autenticación sin contraseña.

No se pueden generar contraseñas automáticamente cuando hay factores sin contraseña disponibles

Si ha configurado su grupo de usuarios para que admita el inicio de sesión OTP por correo o teléfono sin contraseña establecida, no podrá generar una contraseña automáticamente. Para cada usuario que vaya a tener una contraseña, debe establecer una contraseña temporal al crear su perfil.

Los usuarios sin contraseña deben tener valores para todos los atributos obligatorios

Al crear un usuario sin contraseña, la solicitud solo es válida si el usuario proporciona valores para todos los atributos que ha marcado como obligatorios en el grupo de usuarios. Esto se aplica a cualquier atributo obligatorio, no solo a los atributos de número de teléfono y correo electrónico necesarios para la entrega mediante una OTP.

Creación de usuarios que proporcionen los valores de los atributos obligatorios más adelante

Es posible que desee requerir atributos en su grupo de usuarios, pero recopilarlos después de crear los usuarios de forma administrativa, durante la interacción de los usuarios en su aplicación. Los administradores pueden omitir los valores de los atributos obligatorios al crear usuarios con contraseñas temporales. No puede omitir los valores de los atributos obligatorios para los usuarios sin contraseña.

Los usuarios a los que les falten valores en los atributos obligatorios y que tengan una contraseña temporal reciben el comando NEW_PASSWORD_REQUIRED al iniciar sesión por primera vez. A continuación, pueden proporcionar un valor para los atributos obligatorios que faltan en el parámetro `requiredAttributes`. Solo puede crear usuarios con contraseñas y sin los atributos necesarios

si todos los atributos necesarios son mutables. Los usuarios solo pueden completar el inicio de sesión con desafíos `NEW_PASSWORD_REQUIRED` y valores de atributos obligatorios si los atributos necesarios se pueden escribir desde el cliente de aplicación con el que inician sesión.

Cuando establece una contraseña permanente para un usuario creado por el administrador, su estado cambia a `CONFIRMED` y su grupo de usuarios no les pide una nueva contraseña ni los atributos obligatorios la primera vez que inician sesión.

Crear un nuevo usuario en el Consola de administración de AWS

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. Seleccione el menú Métodos de autenticación y busque la 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 el Sign-upmenú y localice el Self-serviceregistro. Seleccione Editar.
4. Elija si desea habilitar el registro automático. Self-registrationse suele utilizar con clientes de aplicaciones públicas que necesitan registrar nuevos usuarios en su 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. Seleccione Save changes (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 el menú Plantillas de mensajes y seleccione Mensaje de verificación, Mensaje de invitación o Mensaje de MFA y seleccione 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. Seleccione Save changes (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. Seleccione Save changes (Guardar cambios).

c. MFA messages (Mensajes MFA)

i. Ingrese una plantilla personalizada de SMS message (Mensaje SMS) para los SMS.

ii. Seleccione Save changes (Guardar cambios).

Creación de un usuario

Creación de 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 las cuentas como administrador, también puedes usar la API AWS CLI 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. Seleccione la pestaña Usuarios y, a continuación, elija 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. Para suprimir el mensaje de invitación, seleccione No enviar una invitación.

> **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 el menú Autenticación 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). La opción para generar una contraseña no está disponible si el inicio de sesión sin contraseña está disponible en el grupo de usuarios. Cualquier contraseña temporal debe cumplir con la política de contraseñas del grupo de usuarios.
8. Seleccione Crear.
9. Elija el menú Usuarios y la entrada 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.

Amazon Cognito crea un grupo de usuarios para cada OIDC, SAMI y proveedor de identidades (IdP) social que añada a su grupo de usuarios. El nombre del grupo tiene el formato `[user pool ID]_[IdP name]`, por ejemplo `us-east-1_EXAMPLE_MYSSO` o `us-east-1_EXAMPLE_Google`. Cada perfil de usuario de IdP único generado automáticamente se agrega automáticamente a este grupo. Los usuarios vinculados no se agregan automáticamente a este grupo, pero puede agregar sus perfiles al grupo en un proceso independiente.

Puede crear y administrar grupos en un grupo de usuarios desde las Consola de administración de AWS 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 elige el grupo con mayor prioridad (menor) y se aplica su función de IAM asociada.
- 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, al grupo con

el valor de prioridad más bajo se le aplica su función de IAM a la `cognito:preferred_role` afirmación del token de identificación del 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.

Crear un grupo nuevo en el Consola de administración de AWS

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 usuario existente.
4. Seleccione el menú Grupos y, a continuación, elija 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 elige el grupo con la prioridad más baja y se aplica su función de IAM asociada. 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

Los grupos de usuarios pueden contener millones de usuarios. Trabajar con un conjunto de datos de este tamaño es un desafío para los administradores. Amazon Cognito cuenta con herramientas para buscar y modificar los perfiles de usuario. Los principales métodos para encontrar usuarios son el menú Usuarios de la consola Amazon Cognito y con. ListUsers De los métodos que recuperan información sobre los usuarios, estas son las opciones que no tienen un impacto en los costos, a diferencia de, por ejemplo, AdminGetUser.

Esta sección de la guía contiene información sobre cómo buscar y actualizar perfiles de usuario en un grupo de usuarios.

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, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija la pestaña Usuarios y 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, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija la pestaña Usuarios y 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.

Habilitación, deshabilitación y eliminación de cuentas de usuario

Puede eliminar los perfiles de usuario no utilizados o, si desea impedir temporalmente el acceso, deshabilitarlos. Los usuarios pueden eliminar sus propias cuentas, pero solo los administradores del grupo de usuarios pueden habilitar y deshabilitar las cuentas de usuario.

Efecto de la eliminación

Los usuarios no pueden iniciar sesión con las cuentas de usuario eliminadas y, para recuperar el acceso, deben registrarse o crearse de nuevo.

Efecto de la deshabilitación de cuentas

Al deshabilitar una cuenta de usuario, Amazon Cognito invalida automáticamente todas las sesiones autenticadas, desactiva la cuenta de usuario para iniciar sesión y revoca sus tokens de acceso y actualización. Amazon Cognito devuelve un mensaje de error `invalid_request` indicando `User is not enabled` cuando un usuario intenta iniciar sesión en una cuenta que usted ha deshabilitado. Este comportamiento no cambia con la configuración de divulgación de la existencia del usuario para el cliente de la aplicación. Puede deshabilitar las cuentas de usuario locales y los perfiles locales de las cuentas de usuario federado. Cuando los usuarios inician sesión con un inicio de sesión administrado o con la clásica interfaz de usuario alojada, usted inhabilita su cuenta y ellos intentan iniciar sesión de nuevo en un navegador con una cookie que mantiene su sesión autenticada, Amazon Cognito los redirige a la página de inicio de sesión.

Efecto de la habilitación de cuentas

Los usuarios pueden iniciar sesión en las cuentas inmediatamente después de que usted las habilite. Las cuentas de usuario están habilitadas de forma predeterminada. Los atributos y las contraseñas de los usuarios siguen siendo los mismos que antes de deshabilitar su cuenta. Los tokens que su aplicación haya revocado, tanto si ha desactivado la cuenta de usuario como si ha revocado por separado el token de actualización, siguen sin ser válidos después de habilitar la cuenta de usuario propietaria del token.

Delete a user account (console)

Cómo eliminar una cuenta de usuario

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 usuario existente.
4. Seleccione el menú Usuarios y seleccione el botón de opciones situado junto al nombre de usuario de la lista.
5. Elija Eliminar.
6. Seleccione Desactivar el acceso de usuarios.
7. Elija Eliminar.

Delete a user account (API)

Los usuarios pueden eliminar sus cuentas mediante la operación de access-token-authorized DeleteUserAPI de autoservicio. A continuación, se muestra un ejemplo de cuerpo de la solicitud `DeleteUser`.

```
{
   "AccessToken": "eyJra456defEXAMPLE"
}
```

Los administradores pueden eliminar las cuentas de usuario mediante la operación de API autorizada por IAM AdminDeleteUser. A continuación, se muestra un ejemplo de cuerpo de la solicitud `AdminDeleteUser`.

```
{
   "Username": "testuser",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Disable a user account (console)

Cómo deshabilitar una cuenta de usuario

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. Seleccione el menú Usuarios y un nombre de usuario para uno de los usuarios en la lista.
5. En la página de datos del usuario, elija Acciones y Desactivar el acceso de usuarios.
6. En el cuadro de diálogo que se creará, seleccione Desactivar.

Disable a user account (API)

Los administradores pueden deshabilitar las cuentas de usuario mediante la operación de AdminDisableUserAPI autorizada por IAM. A continuación, se muestra un ejemplo de cuerpo de la solicitud `AdminDisableUser`.

```
{
   "Username": "testuser",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Enable a user account (console)

Cómo habilitar una cuenta de usuario

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. Seleccione el menú Usuarios y un nombre de usuario para uno de los usuarios en la lista.
5. En la página de datos del usuario, elija Acciones y Habilitar el acceso de usuarios.
6. En el cuadro de diálogo que se creará, seleccione Habilitar.

Enable a user account (API)

Los administradores pueden habilitar las cuentas de usuario con la operación de AdminEnableUserAPI autorizada por IAM. A continuación, se muestra un ejemplo de cuerpo de la solicitud `AdminEnableUser`.

```
{
   "Username": "testuser",
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Búsqueda de atributos de usuario

Si ya ha creado un grupo de usuarios, puede realizar búsquedas desde el panel Users (Usuarios) de la Consola de administración de AWS. También puede usar la ListUsers API de Amazon Cognito, que acepta un parámetro de 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
- name
- 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 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 sobre el filtrado del lado del servidor y del lado del cliente, consulte Filtrar los resultados en la Guía del usuario. AWS CLI AWS Command Line Interface

Búsqueda de usuarios con Consola de administración de AWS

Si ya ha creado un grupo de usuarios, puede realizar búsquedas desde el panel Users (Usuarios) de la Consola de administración de AWS.

Consola de administración de AWS las búsquedas son siempre búsquedas con prefijos («comienza por»).

Para buscar un usuario en la consola de Amazon Cognito

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 en la lista un usuario existente.
4. Seleccione el menú Usuarios e introduzca el nombre de 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 desde su aplicación, utilice la API de Amazon CognitoListUsers . 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"
}
```

Contraseñas, recuperación de contraseñas y políticas de contraseñas

Todos los usuarios que inician sesión en un grupo de usuarios, incluso los usuarios federados, tienen contraseñas asignadas a sus perfiles de usuario. Los usuarios locales y los usuarios vinculados deben proporcionar una contraseña al iniciar sesión. Los usuarios federados no utilizan contraseñas de grupos de usuarios, sino que inician sesión con su proveedor de identidades (IdP). Puede permitir a los usuarios restablecer sus propias contraseñas, restablecer o cambiar las contraseñas como administradores y establecer políticas relativas a la complejidad y el historial de las contraseñas.

Amazon Cognito no almacena las contraseñas de los usuarios en texto sin formato. En su lugar, almacena un hash de la contraseña de cada usuario con un salt específico para cada usuario. Por

este motivo, no puede recuperar las contraseñas existentes de los perfiles de usuario de los grupos de usuarios. Como práctica recomendada, no guarde en ningún lugar contraseñas de usuario en texto sin formato. Restablezca las contraseñas cuando los usuarios las olviden.

Restablecimiento y recuperación de la contraseña

Los usuarios se olvidan de las contraseñas. Puede que quiera darles la posibilidad de que ellos mismos restablezcan su propia contraseña o puede que quiera exigir que un administrador restablezca la contraseña en su lugar. Los grupos de usuarios de Amazon Cognito cuentan con opciones para ambos modelos. En esta parte de la guía se aborda la configuración del grupo de usuarios y las operaciones de API necesarias para restablecer las contraseñas.

El funcionamiento ForgotPasswordde la API y la opción de inicio de sesión gestionado ¿Ha olvidado su contraseña? envía a los usuarios un código que, cuando confirmen que tienen el código correcto, les da la oportunidad de establecer una nueva contraseña ConfirmForgotPassword. Se trata del modelo de autoservicio de recuperación de contraseñas.

Recuperación de usuarios no verificados

Puede enviar mensajes de recuperación a los usuarios que hayan verificado su dirección de correo electrónico o número de teléfono. Si no tienen un correo electrónico o un teléfono de recuperación confirmados, el administrador del grupo de usuarios puede marcar su dirección de correo electrónico o número de teléfono como verificados. Edite los Atributos de usuario en la consola de Amazon Cognito y seleccione la casilla situada junto a Marque el número de teléfono como verificado o Marque la dirección de email como verificada. También puedes establecer `email_verified` o como `phone_number_verified` true en una AdminUpdateUserAttributessolicitud. En el caso de los nuevos usuarios, la operación de la ResendConfirmationCodeAPI envía un código nuevo a su dirección de correo electrónico o número de teléfono y estos pueden completar la confirmación y la verificación mediante el autoservicio.

Restablecimiento de las contraseñas como administrador

Las operaciones de AdminResetUserPasswordAPI AdminSetUserPasswordy las operaciones son los métodos de restablecimiento de contraseñas iniciados por el administrador. `AdminSetUserPassword`destablece una contraseña temporal o permanente y `AdminResetUserPassword` envía a los usuarios un código para restablecer la contraseña de la misma manera que. `ForgotPassword`

Configuración del restablecimiento y la recuperación de contraseñas

Amazon Cognito selecciona automáticamente las opciones de recuperación de la cuenta entre los atributos necesarios y las opciones de inicio de sesión que elija al crear un grupo de usuarios en la consola. Puede modificar esta configuración predeterminada.

El método de MFA preferido del usuario influye en los métodos que este pueda utilizar para recuperar la contraseña. Los usuarios cuya MFA preferida se realice por mensaje de correo electrónico no pueden recibir un código de restablecimiento de contraseña por correo electrónico. Los usuarios cuya MFA preferida se realice por mensaje SMS no pueden recibir un código de restablecimiento de contraseña por SMS.

La configuración de la recuperación de contraseñas debe ofrecer una opción alternativa para cuando el usuario no pueda utilizar el método de restablecimiento de contraseña preferido. Por ejemplo, puede darse el caso de que sus mecanismos de recuperación tengan el correo electrónico como primera prioridad y la MFA de correo electrónico puede ser una opción en el grupo de usuarios. En este caso, añada la recuperación de SMS-message cuentas como segunda opción o utilice las operaciones administrativas de la API para restablecer las contraseñas de esos usuarios.

Amazon Cognito responde a las solicitudes de restablecimiento de contraseñas de usuarios que no disponen de un método de recuperación válido con una respuesta de error `InvalidParameterException`.

> **Note**
>
> Los usuarios no pueden recibir códigos de MFA y de restablecimiento de contraseña en la misma dirección de correo electrónico o número de teléfono. Si usan contraseñas de un solo uso (OTP) desde los mensajes de correo electrónico para la MFA, deben usar mensajes SMS para recuperar la cuenta. Si usan OTP desde los mensajes SMS para la MFA, deben usar mensajes de correo electrónico para recuperar la cuenta. En los grupos de usuarios con MFA, es posible que los usuarios no puedan completar la recuperación automática de contraseñas si tienen atributos para su dirección de correo electrónico, pero no un número de teléfono, o si su número de teléfono no tiene una dirección de correo electrónico.
> Para evitar que los usuarios no puedan restablecer sus contraseñas en los grupos de usuarios con esta configuración, defina los atributos `email` y `phone_number` según sea necesario. Si lo prefiere, puede configurar procesos que siempre recopilen y establezcan esos atributos cuando los usuarios se registren o cuando los administradores creen perfiles de usuario. Cuando los usuarios tienen ambos atributos, Amazon Cognito envía

automáticamente códigos de restablecimiento de contraseñas al destino que no sea el factor de MFA del usuario.

El siguiente procedimiento configura la recuperación automática de cuentas en un grupo de usuarios.

Configure self-service password reset (API/SDK)

El `AccountRecoverySetting` parámetro es el parámetro del grupo de usuarios que establece los métodos que los usuarios pueden usar para recuperar su contraseña en las solicitudes de ForgotPasswordAPI o cuando seleccionan ¿Ha olvidado su contraseña? en un inicio de sesión gestionado. `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 del 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 teniendo en cuenta que 1 representa la prioridad más alta. Amazon Cognito envía una verificación solo a uno de los métodos especificados. En el siguiente ejemplo, `AccountRecoverySetting` establece las direcciones de correo electrónico como destino principal de los códigos de recuperación de cuentas, y se recurre a los mensajes SMS si el usuario no tiene un atributo de dirección de correo electrónico.

```
"AccountRecoverySetting": {
   "RecoveryMechanisms": [
      {
         "Name": "verified_email",
         "Priority": 1
      },
      {
         "Name": "verified_phone_number",
         "Priority": 2
      }
   ]
}
```

El valor `admin_only` desactiva la recuperación automática de cuentas y les pide a los usuarios que se pongan en contacto con su administrador para restablecer la contraseña. No se puede utilizar `admin_only` con ningún otro mecanismo de recuperación de la cuenta. El siguiente e

```
"AccountRecoverySetting": {
   "RecoveryMechanisms": [
      {
         "Name": "admin_only",
         "Priority": 1
      }
   ]
}
```

Si no especifica `AccountRecoverySetting`, Amazon Cognito envía primero el código de recuperación a un número de teléfono verificado y a una dirección de correo electrónico verificada si los usuarios no tienen un atributo de número de teléfono.

Para obtener más información sobre `AccountRecoverySetting`, consulte CreateUserPool y UpdateUserPool.

Configure self-service password reset (console)

Configura las opciones de recuperación de cuentas y restablecimiento de contraseñas desde el Sign-inmenú de tu grupo de usuarios.

Cómo configurar la recuperación de cuentas de usuario

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. Seleccione el menú Sign-in. Busque la Recuperación de cuenta de usuario y seleccione Editar
5. Para permitir a los usuarios restablecer sus propias contraseñas, seleccione Habilitar la recuperación automática de cuentas.
6. Configure el método de entrega de los códigos de recuperación de contraseñas que el grupo de usuarios envía a los usuarios. En Método de entrega para los mensajes de recuperación de cuentas de usuario, seleccione una opción disponible. Como práctica recomendada, elija una opción que tenga un método secundario para el envío de mensajes, como Correo electrónico si está disponible, de lo contrario, SMS. Con un método de entrega secundario, Amazon Cognito puede enviar códigos a los usuarios de una forma que les obligue a utilizar un medio de restablecimiento de contraseñas diferente al de la MFA.
7. Seleccione Guardar cambios.

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.

Adición de requisitos de contraseña para los grupos de usuarios

Las contraseñas complejas y seguras son una práctica recomendada de seguridad para los grupos de usuarios. Las contraseñas poco seguras, especialmente en el caso de las aplicaciones que están abiertas a Internet, pueden exponer las credenciales de los usuarios a sistemas que las adivinen e intenten acceder a los datos. Cuanto más compleja sea la contraseña, más difícil será adivinarla. Amazon Cognito cuenta con herramientas adicionales para los administradores que quieren priorizar la seguridad, como la protección contra amenazas y las ACL web de AWS WAF, pero la política de contraseñas que implante será el elemento fundamental de la seguridad en el 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 de restablecimiento de las contraseñas de los usuarios en un sistema externo. Con un registro externo de la antigüedad de las contraseñas, su aplicación o un desencadenador de Lambda pueden buscar la antigüedad de la contraseña de un usuario y requerir que se restablezca después de un período de tiempo determinado.

Puede configurar el grupo de usuarios para que la complejidad mínima de la contraseña cumpla los estándares de seguridad. Las contraseñas complejas deben tener una longitud mínima de ocho caracteres. También incluyen una combinación de caracteres numéricos, especiales y en mayúsculas.

Con los niveles de características Essentials o Plus, también puede establecer una política de reutilización de contraseñas. Puede impedir que un usuario restablezca su contraseña por una nueva que coincida con la contraseña actual o cualquiera de las 23 contraseñas anteriores adicionales, de un total de 24 como máximo.

> **Note**
>
> Amazon Cognito no aplica su política de contraseñas cuando importa usuarios con hashes de contraseña. Valide la seguridad de la contraseña en su sistema de origen antes de generar hashes para la importación.

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 al menú Métodos de autenticación.
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 exigir al menos uno de los siguientes caracteres en las contraseñas. Una vez que Amazon Cognito compruebe que las contraseñas contienen los caracteres mínimos necesarios, 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.

     ```
     ^ $ * . [ ] { } ( ) ? " ! @ # % & / \ , > < ' : ; | _ ~ ` = + -
     ```

   - Non-leading, caracteres de espacio sin espacios finales.
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 TemporaryPasswordValidityDays su solicitud CreateUserPoolde UpdateUserPoolAPI.

7. Establezca un valor para Impedir el uso de contraseñas anteriores, si está disponible. Para utilizar esta característica, elija el nivel de características Essentials o Plus en el grupo de usuarios. El valor de este parámetro es el número de contraseñas anteriores que no deben ser iguales a la nueva contraseña cuando un usuario restablece esta última.

Para restablecer el acceso de una cuenta de usuario caducada, realice una de las siguientes acciones:

- Envíe una nueva contraseña temporal y restablezca el período de caducidad con una solicitud de AdminCreateUserAPI que se haya `MessageAction` establecido en. `RESEND`
- Elimine el perfil de usuario y cree uno nuevo.
- Genera un nuevo código de confirmación en una solicitud de AdminResetUserPasswordAPI.

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 enfoque, los usuarios pueden seguir usando sus contraseñas actuales y no tienen que restablecerlas después de la migración a su 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.

Más recursos

- Approaches for migrating users to Amazon Cognito user pools
- AWS Re:inForce 2023 - Migración a Amazon Cognito

Temas

- Importación de usuarios con un desencadenador de Lambda para la migración de usuarios
- Importación de usuarios en grupos de usuarios desde un archivo CSV

Importación 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).

Sign-in proceso

1. El usuario abre la aplicación e inicia sesión con la API de grupos de usuarios de Amazon Cognito o el inicio de sesión administrado. 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 tu aplicación tiene una interfaz de inicio de sesión personalizada que creaste con un AWS SDK, tu aplicación debe usar InitiateAutho AdminInitiateAuthcon el `ADMIN_USER_PASSWORD_AUTH` flujo `USER_PASSWORD_AUTH` o. 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 del usuario como alias en su grupo de usuarios. Si el usuario no existe, Amazon Cognito envía los parámetros, incluidos el nombre de usuario y la contraseña, a su Migración del desencadenador de Lambda del usuario función.
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 `username` parámetro 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 desinfecta los registros, las contraseñas aparecen en CloudWatch los registros.

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 su aplicación, solicite al usuario su código de confirmación y una nueva contraseña y, a continuación, envíe esa información a Amazon Cognito en una solicitud de ConfirmForgotPasswordAPI. Puede también utilizar páginas integradas para el flujo de contraseña olvidada en el inicio de sesión administrado.

Recursos adicionales

- Approaches for migrating users to Amazon Cognito user pools

Importación de usuarios en grupos de usuarios desde un archivo CSV

Si dispone de un almacén de identidades externo y tiene tiempo de preparar su grupo de usuarios para los nuevos usuarios locales, la importación masiva de usuarios desde un archivo de valores separados por comas (CSV) puede ser una opción económica y que requiere poco esfuerzo para migrar a un grupo de usuarios de Amazon Cognito. La importación de un archivo CSV es un proceso que consiste en descargar y rellenar un archivo de plantilla y, a continuación, ponerlo a disposición del grupo de usuarios en un trabajo de importación. Puede utilizar una importación de CSV para crear rápidamente usuarios de prueba. También puede rellenar el archivo mediante programación con solicitudes de API de lectura al almacén de identidades externo y, a continuación, analizar sus detalles y atributos para convertirlas en operaciones de escritura en el archivo.

De forma predeterminada, el proceso de importación establece valores para todos los atributos del usuario, excepto la contraseña. Esto significa que sus usuarios deben cambiar de contraseña la primera vez que inicien sesión. Los usuarios se encuentran en estado `RESET_REQUIRED` cuando se importan con este método.

Como alternativa, puede importar los usuarios con los hashes de contraseña existentes. Cuando especifica un algoritmo de cifrado de contraseñas durante la creación del trabajo de importación e incluye códigos hash de contraseñas en el archivo CSV, Amazon Cognito importa los usuarios

con sus contraseñas existentes. Estos usuarios se crean con un `CONFIRMED` estado y pueden iniciar sesión inmediatamente sin necesidad de restablecer sus contraseñas. Para obtener más información, consulte Importación de usuarios con códigos hash de contraseña.

La forma más sencilla de importar usuarios desde un CSV es activar el inicio de sesión sin contraseña en su grupo de usuarios. Con los atributos de dirección de correo electrónico y número de teléfono y la configuración correcta del grupo de usuarios, los usuarios pueden iniciar sesión con contraseñas de un solo uso (OTP) por correo electrónico o SMS inmediatamente después de completar el trabajo de importación. Para obtener más información, consulte Obligación de que los usuarios importados restablezcan sus contraseñas.

También puede establecer las contraseñas de sus usuarios con una solicitud a la API AdminSetUserPassword que establezca el parámetro `Permanent` en `true`. La importación de CSV no contribuye a los usuarios activos mensuales (MAU) facturados de su grupo de usuarios. Sin embargo, las operaciones de restablecimiento de contraseña generan MAU. Para administrar los costos que supone importar un gran número de usuarios con contraseña que podrían no estar activos inmediatamente, configure la aplicación para que solicite a los usuarios que introduzcan una nueva contraseña cuando inicien sesión y reciban el desafío `RESET_REQUIRED`.

> **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.

Pasos para crear un trabajo de importación de usuarios

1. Cree un rol de Amazon CloudWatch Logs en la consola AWS Identity and Access Management (IAM).
2. Cree el archivo .csv de importación de usuarios.
3. Cree y ejecute el trabajo de importación de usuarios. Si lo desea, especifique un algoritmo de cifrado de contraseñas para importar los usuarios con los códigos hash de contraseñas existentes.
4. Cargue el archivo .csv de importación de usuarios.
5. Inicie y ejecute el trabajo de importación de usuarios.
6. Se utiliza CloudWatch para comprobar el registro de eventos.
7. Si no has importado los códigos hash de las contraseñas, pide a los usuarios importados que restablezcan sus contraseñas.

Más recursos

- Arquitectura de referencia de exportación de perfiles de usuario de Cognito para exportar cuentas de usuario entre grupos de usuarios

Temas

- Crear el rol de CloudWatch IAM de registros
- Creación del archivo CSV de importación de usuarios
- Creación y ejecución del trabajo de importación del grupo de usuarios de Amazon Cognito
- Ver los resultados de importación del grupo de usuarios en la CloudWatch consola
- Obligación de que los usuarios importados restablezcan sus contraseñas
- Importación de usuarios con códigos hash de contraseña

Crear el rol de CloudWatch IAM de registros

Si utiliza la CLI o la API de Amazon Cognito, debe crear un rol de CloudWatch IAM. El siguiente procedimiento describe cómo crear un rol de IAM que Amazon Cognito pueda usar para escribir los resultados del trabajo CloudWatch de importación en 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 CloudWatch de registros para la importación de grupos de usuarios (AWS CLI, API)

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala 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. *REGION*Reemplácelo por el Región de AWS de su grupo de usuarios, por ejemplo`us-east-1`. *ACCOUNT*Sustitúyalo por tu Cuenta de AWS ID, por ejemplo`111122223333`.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333: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 Relaciones de confianza.
5. Elija 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:

JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "cognito-idp.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    }
```

7. Elija Actualizar política.
8. Apunte el ARN del rol . Proporcionará el ARN cuando cree su trabajo de importación.

Como alternativa, puede crear el mismo rol con la AWS CLI. Los siguientes comandos crean una función que Amazon Cognito puede asumir, adjuntan una política que permita escribir en CloudWatch Logs y devuelven la función ARN. *REGION*Sustitúyalo por el Región de AWS de su grupo de usuarios y por su *ACCOUNT* Cuenta de AWS ID.

Para crear la función de IAM de CloudWatch registros para la importación del grupo de usuarios (AWS CLI)

1. Cree el rol con una política de confianza que permita a Amazon Cognito asumirlo.

```
aws iam create-role \
    --role-name CognitoUserImportRole \
    --assume-role-policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": { "Service": "cognito-idp.amazonaws.com" },
                "Action": "sts:AssumeRole"
            }
        ]
```

```
    }'
```

2. Adjunte una política que permita al rol escribir los resultados de la importación en los CloudWatch registros.

```
aws iam put-role-policy \
    --role-name CognitoUserImportRole \
    --policy-name CognitoUserImportLogsPolicy \
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "logs:CreateLogGroup",
                    "logs:CreateLogStream",
                    "logs:PutLogEvents",
                    "logs:DescribeLogStreams"
                ],
                "Resource": "arn:aws:logs:REGION:ACCOUNT:log-group:/aws/cognito/*"
            }
        ]
    }'
```

3. Recupera el ARN del rol para usarlo *ROLE_ARN* cuando crees tu trabajo de importación.

```
aws iam get-role \
    --role-name CognitoUserImportRole \
    --query 'Role.Arn' --output text
```

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. 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 en la lista un usuario existente.
4. Seleccione el menú Usuarios.
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.

Descargar el encabezado del archivo CSV (AWS CLI)

Para obtener una lista de los encabezados correctos, en el menú Usuarios, en Importar usuarios, seleccione Crear trabajo de importación. En el cuadro de diálogo siguiente, seleccione el enlace `template.csv` para descargar un archivo de plantilla con los atributos de su grupo de usuarios.

También puede ejecutar el siguiente comando CLI, donde *`USER_POOL_ID`* se encuentra el identificador del grupo de usuarios al que va a importar 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 Uso de atributos de usuario.

- 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
  - 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 tener un UTF-8 formato 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 de la fecha de nacimiento, si están presentes, deben estar en ese formato *mm/dd/yyyy*. 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 debe ajustarse a los requisitos de la MFA de su grupo de usuarios. Si ha establecido que la autenticación multifactor (MFA) sea obligatoria en su grupo de usuarios, este campo debe ser `true`, o dejarse en blanco, para todos los usuarios. Si ha desactivado la MFA, este campo debe ser `false`, o dejarse en blanco, para todos los usuarios. Un valor en

blanco establece el MFA-enabled estado de los usuarios importados en el estado requerido por el grupo de usuarios. Puede importar usuarios de un grupo de MFA-required usuarios sin un factor de MFA válido, independientemente de si establece un `cognito:mfa_enabled` valor. Los usuarios de este estado tienen la MFA activa, pero no pueden iniciar sesión hasta que configuren un atributo de correo electrónico, un atributo de número de teléfono o un TOTP, y esa configuración sea un factor de MFA válido en su grupo de usuarios.

- 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**.
- Se recorta cualquier espacio en blanco al principio o al final del valor de un atributo.

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 el 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. 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 en la lista un usuario existente.
4. Seleccione el menú Usuarios.
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 crea 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 Crear el rol de CloudWatch IAM de registros.
8. En Cargar CSV, seleccione Elegir archivo y adjunte el archivo CSV que ha preparado.
9. 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.
10. Si ha creado el trabajo pero no lo ha iniciado, puede iniciarlo más adelante. En el menú Usuarios, en Importar usuarios, elija su trabajo de importación y, a continuación, seleccione Iniciar. También puede enviar una solicitud de StartUserImportJobAPI desde un AWS SDK.
11. Supervise el progreso de su trabajo de importación de usuarios en la pestaña Usuarios, en Importar usuarios. Si su trabajo no se realiza correctamente, puede seleccionar el valor Status (Estado). Para obtener más información, selecciona Ver los CloudWatch registros para obtener más información y consulta cualquier problema en la consola de CloudWatch registros.

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 ejecutando el siguiente comando CLI, donde *JOB_NAME* es el nombre que va a elegir para el trabajo, *USER_POOL_ID* es el ID del grupo de usuarios del grupo de usuarios al que se agregan los nuevos usuarios y *ROLE_ARN* es el ARN de rol que recibió: Crear el rol de CloudWatch IAM de registros

```
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 *PRE_SIGNED_URL* valor devuelto en la respuesta es válido durante 15 minutos. Transcurrido ese tiempo, caduca y debes crear un nuevo trabajo de importación de usuarios para obtener una nueva URL.

Example respuesta:

```
{
    "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. Sus grupos de usuarios no conservan la información de los archivos de importación después de ejecutar los trabajos de importación. Cuando se completen o caduquen, Amazon Cognito eliminará el archivo CSV cargado.

Descripción de un trabajo de importación de usuarios

Para obtener una descripción del trabajo de importación de usuarios, utilice el siguiente comando, donde *`USER_POOL_ID`* está el ID del grupo de usuarios y *`JOB_ID`* el ID del trabajo que se devolvió 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 cargó el archivo CSV. *ROLE_ARN*Es el ARN del rol de CloudWatch registros que recibió al crear 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 *PAGINATION_TOKEN* cadena que sigue al segundo trabajo indica que hay resultados adicionales para este comando de lista. 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
    }
}
```

Ver los resultados de importación del grupo de usuarios en la CloudWatch consola

Puedes ver los resultados de tu trabajo de importación en la CloudWatch consola de Amazon.

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. Inicia sesión en Consola de administración de AWS y abre la CloudWatch consola 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 del registro 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 importados correctamente tienen 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:

- El rol de Amazon CloudWatch Logs no se puede asumir, 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

Si su grupo de usuarios solo ofrece el inicio de sesión basado en contraseñas, los usuarios deberán restablecer sus contraseñas después de importarlas. La primera vez que inicien sesión, podrán introducir cualquier contraseña. Amazon Cognito les pide que introduzcan una nueva contraseña en la respuesta de la API a la solicitud de inicio de sesión de su aplicación.

Si su grupo de usuarios tiene factores de autenticación sin contraseña, Amazon Cognito utiliza de forma predeterminada los de los usuarios importados. No se les pide una contraseña nueva y pueden iniciar sesión inmediatamente con una OTP por correo o SMS sin contraseña establecida. También puede pedir a los usuarios que establezcan una contraseña para que puedan completar otros métodos de inicio de sesión, como la combinación de nombre de usuario y contraseña o el uso de una clave de acceso. Tras la importación del usuario, se aplican las siguientes condiciones al inicio de sesión sin contraseña.

1. Debe importar los usuarios con un atributo que corresponda a un factor de inicio de sesión sin contraseña disponible. Si los usuarios pueden iniciar sesión con una dirección de correo electrónico, debe importar un atributo `email`. Si es un número de teléfono, debe importar un atributo `phone_number`. Si son ambas cosas, importe un valor para cualquiera de los atributos.
2. Normalmente, los usuarios importan en un estado `RESET_REQUIRED` en el que deben restablecer su contraseña. Si se importan con la posibilidad de iniciar sesión sin contraseña, Amazon Cognito establece su estado en `CONFIRMED`.

Para obtener más información sobre la autenticación sin contraseña, lo que incluye cómo configurarla y cómo crear el flujo de autenticación en su aplicación, consulte Autenticación con grupos de usuarios de Amazon Cognito.

El siguiente procedimiento describe la experiencia del usuario en un mecanismo de inicio de sesión personalizado con usuarios locales en un `RESET_REQUIRED` después de importar un archivo CSV. Si sus usuarios inician sesión con un inicio de sesión administrado, pídales que seleccionen la opción ¿Ha olvidado su contraseña?, que miren el código en su correo electrónico o mensaje de texto y que establezcan una contraseña.

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.

Importación de usuarios con códigos hash de contraseña

Al migrar usuarios de un sistema de autenticación existente a Amazon Cognito, puede importar los hashes de sus contraseñas junto con sus atributos de usuario. A continuación, sus usuarios pueden iniciar sesión inmediatamente con sus contraseñas existentes, sin necesidad de restablecer la contraseña. Cuando los usuarios inician sesión por primera vez, Amazon Cognito migra sus credenciales de forma transparente al sistema de autenticación nativo de Amazon Cognito.

Cómo funciona la importación de hash de contraseñas

Con la importación del hash de contraseñas, puede migrar a los usuarios y, al mismo tiempo, conservar su capacidad de iniciar sesión con sus contraseñas existentes. El proceso de importación funciona de la siguiente manera:

1. Cree un trabajo de importación y especifique el algoritmo de cifrado de contraseñas que el sistema de origen utilizó para crear los códigos hash de las contraseñas.
2. Incluyes la `password_hash` columna en el archivo CSV con el valor de hash de la contraseña de cada usuario.
3. Amazon Cognito importa los usuarios con sus códigos hash de contraseña y los crea con un estado. `CONFIRMED` Los usuarios pueden iniciar sesión inmediatamente.
4. Cuando un usuario inicia sesión por primera vez, Amazon Cognito verifica su contraseña con el hash importado. Tras la verificación correcta, Amazon Cognito migra las credenciales del usuario al sistema de autenticación nativo.
5. Todos los inicios de sesión posteriores utilizan el sistema de autenticación nativo de Amazon Cognito.

> **Important**
>
> Todos los hashes de contraseñas de un único trabajo de importación deben utilizar el mismo algoritmo. El algoritmo se especifica a nivel de trabajo al crear el trabajo de importación.

> **Note**
>
> Hasta que un usuario complete su primer inicio de sesión y Amazon Cognito migre sus credenciales, no podrá usar la autenticación de contraseña remota segura (SRP) para ese

usuario. Use `USER_PASSWORD_AUTH` o `ADMIN_USER_PASSWORD_AUTH` flujos para los usuarios con hashes de contraseña importados que aún no hayan iniciado sesión.

Algoritmos de cifrado de contraseñas compatibles

Amazon Cognito admite los siguientes algoritmos de hash de contraseñas para la importación:

`BCRYPT`

La función de hash adaptativo bcrypt. Amazon Cognito extrae todos los parámetros necesarios (sal, factor de coste) de la cadena de hash.

Formato: `$2<a/b/x/y>$[cost]$[22-char salt][31-char hash]`

Ejemplo: `$2b$10$CtA.Rcu/szzn9U00wpUjOuN3vrgJRZycv4aOzcP3GzqzO8UDPEFq6`

Factor de coste máximo: 12

`SCRYPT`

La función de derivación de claves basada en contraseñas de Scrypt. Amazon Cognito extrae todos los parámetros necesarios de la cadena hash.

Formato: `N$r$p$hexSalt$hexHash`

Ejemplo: `65536$8$1$304dbaef7c5e828dc19c98f0600d18fe $4f69c498c12cd102d057356facf8d77e8d42407090491ea32c5b038f5a18c099`

Parámetros máximos: N (CPU/memory coste) = 65536, r (tamaño del bloque) = 8, p (paralelismo) = 1

`ARGON2ID`

El algoritmo de cifrado de contraseñas Argon2ID. Amazon Cognito extrae todos los parámetros necesarios de la cadena hash.

Formato: `$argon2id$v=N$m=M,t=T,p=P$salt$hash`

Ejemplo: `$argon2id$v=19$m=19456,t=2,p=1$ko/G5o1ms+ML08P95sQ8DA $AkVbvWSOqz7Hs3qthhWKxicOWnGLN+MBmpwc3emi5VA`

Parámetros máximos: m (memoria en KiB) = 19456, t (iteraciones) = 2, p (paralelismo) = 1

`PBKDF2_SHA256`

Password-Based Función de derivación de claves 2 con. SHA-256 Amazon Cognito extrae todos los parámetros necesarios de la cadena hash.

Formato: `$pbkdf2-sha256$iterations$salt$hash`

Ejemplo: `$pbkdf2-sha256$600000$1XZlmwLQ2hhM3JYuCPiArQ$Pfheg9Zi/v5lXU4yyLA0WFUYEd/rlaVbzrM9oMD6IrA`

Número máximo de iteraciones: 600000

> **Note**
>
> Todos los algoritmos compatibles se autodescriben, lo que significa que Amazon Cognito puede extraer todos los parámetros necesarios (como la sal, el factor de coste y las iteraciones) directamente de la cadena hash. Solo tiene que especificar el nombre del algoritmo al crear el trabajo de importación.

Si el hash de la contraseña tiene valores de parámetros que superan los límites máximos indicados anteriormente, la importación no se realizará para ese usuario. Revise la configuración de cifrado de contraseñas del sistema de origen antes de realizar la importación para garantizar la compatibilidad.

Añadir códigos hash de contraseñas a tu archivo CSV

Al descargar la plantilla CSV para importarla por el usuario, la plantilla incluye una `password_hash` columna. Rellene esta columna con los valores hash de las contraseñas de los usuarios que desee importar con sus contraseñas existentes.

Reglas de formato CSV para la importación del hash de contraseñas

- La `password_hash` columna es obligatoria cuando el trabajo de importación especifica un algoritmo de cifrado de contraseñas. Si deja el valor vacío para un usuario específico, Amazon Cognito importa ese usuario sin contraseña y establece el estado para el `RESET_REQUIRED` usuario.
- Si el trabajo de importación especifica un algoritmo de cifrado de contraseñas, pero el archivo CSV no incluye ninguna `password_hash` columna, se produce un error en el trabajo.

- Si el valor hash de una contraseña tiene un formato incorrecto o no coincide con el formato esperado para el algoritmo especificado, se producirá un error en la importación de ese usuario. Amazon Cognito no crea el usuario y registra el error en los registros de Amazon CloudWatch Logs.
- Los valores hash de las contraseñas distinguen mayúsculas de minúsculas. Asegúrese de que coincidan con el formato exacto que espera el algoritmo.

Example Ejemplo de CSV con códigos hash de contraseña

El siguiente ejemplo muestra un archivo CSV que importa usuarios con hashes de contraseña bcrypt:

```
cognito:username,email,email_verified,phone_number,phone_number_verified,password_hash
alejandro_rosalez,alejandro_rosalez@example.com,TRUE,+12345550100,TRUE,$2b$10$CtA.Rcu/
szzn9U00wpUjOuN3vrgJRZycv4aOzcP3GzqzO8UDPEFq6
mary_major,mary_major@example.com,TRUE,+12345550199,TRUE,$2b$10$CtA.Rcu/
szzn9U00wpUjOuN3vrgJRZycv4aOzcP3GzqzO8UDPEFq6
```

Crear un trabajo de importación con hashes de contraseña (AWS CLI)

Para importar usuarios con códigos hash de contraseña, debe especificar el algoritmo de cifrado de contraseñas al crear el trabajo de importación. Todos los usuarios del trabajo de importación deben usar el mismo algoritmo.

Utilice el siguiente AWS Command Line Interface comando para crear un trabajo de importación con hashes de contraseña. El `--password-hashing-algorithm` parámetro especifica el algoritmo utilizado para crear los hashes de contraseña en el archivo CSV.

```
aws cognito-idp create-user-import-job \
    --job-name "JOB_NAME" \
    --user-pool-id "USER_POOL_ID" \
    --cloud-watch-logs-role-arn "ROLE_ARN" \
    --password-hashing-algorithm BCRYPT
```

*BCRYPT*Sustitúyalo por uno de los algoritmos compatibles: `BCRYPT`,`SCRYPT`,`ARGON2ID`, o`PBKDF2_SHA256`.

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",
        "PasswordHashingAlgorithm": "BCRYPT",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
}
```

La respuesta incluye el `PasswordHashingAlgorithm` campo que confirma el algoritmo que especificó. Cargue el archivo CSV en la URL prefirmada e inicie el trabajo tal y como se describe enCarga del archivo CSV.

Experiencia de usuario tras la importación del hash de la contraseña

Los usuarios importados con códigos hash de contraseña tienen la siguiente experiencia:

- Inicio de sesión inmediato: los usuarios pueden iniciar sesión con sus contraseñas existentes inmediatamente después de la importación. Los usuarios no reciben ninguna solicitud para restablecer sus contraseñas.
- Estado del usuario: Amazon Cognito crea estos usuarios con un `CONFIRMED` estado en lugar de. `RESET_REQUIRED`
- Migración transparente: cuando los usuarios inician sesión por primera vez, Amazon Cognito verifica su contraseña con el hash importado. Tras la verificación correcta, Amazon Cognito migra sus credenciales al sistema de autenticación nativo. Todos los inicios de sesión posteriores utilizan el sistema nativo.
- Latencia de inicio de sesión inicial: el primer inicio de sesión de los usuarios con códigos hash de contraseña importados puede tardar un poco más que los inicios de sesión posteriores. Amazon Cognito debe verificar la contraseña con el hash importado y migrar las credenciales al sistema de autenticación nativo.
- Restablecimiento de contraseña: si un usuario restablece su contraseña antes de iniciar sesión por primera vez, el hash de contraseña importado se sustituye por una nueva contraseña mediante el sistema de autenticación nativo de Amazon Cognito.

Solución de errores al importar el hash de contraseñas

Si se produce un error al importar un hash de contraseña para un usuario, Amazon Cognito registra el error en sus registros de Amazon CloudWatch Logs. Los escenarios de error más comunes incluyen:

Hash mal formado

El hash de la contraseña no coincide con el formato esperado para el algoritmo especificado. Compruebe que el formato hash es correcto y coincide con uno de los formatos enumerados enAlgoritmos de cifrado de contraseñas compatibles.

Parámetro fuera de los límites

El hash de la contraseña contiene valores de parámetros (como el factor de coste o las iteraciones) que superan los valores máximos permitidos. Para los usuarios con hashes incompatibles, elija una de las siguientes opciones:

- Importe los usuarios sin un hash de contraseña dejando el `password_hash` campo vacío. Amazon Cognito los establece en ese `RESET_REQUIRED` estado y deben restablecer sus contraseñas la primera vez que inicien sesión.
- Utilice un activador Lambda de migración de usuarios para migrar esos usuarios al iniciar sesión en lugar de hacerlo mediante la importación de CSV.
- Antes de retirar el sistema de origen, vuelva a codificar las contraseñas con parámetros compatibles cuando los usuarios inicien sesión en el sistema actual.

El algoritmo no coincide

El hash de la contraseña se creó con un algoritmo diferente al especificado en el trabajo de importación. Todos los hashes de un solo trabajo de importación deben usar el mismo algoritmo. Si tiene usuarios con algoritmos diferentes, cree trabajos de importación independientes para cada algoritmo.

Falta la columna password_hash

Especificaste un algoritmo de cifrado de contraseñas al crear el trabajo, pero tu archivo CSV no incluye la columna. `password_hash` Agrega la columna a tu archivo CSV o crea un nuevo trabajo sin especificar un algoritmo.

Cuando se produce un error al importar un hash de contraseña para un usuario, Amazon Cognito no crea el usuario. Los demás usuarios del trabajo de importación se siguen procesando. Revisa

tus CloudWatch registros de Amazon Logs para identificar qué usuarios fallaron y el motivo de cada error.

Consideraciones de seguridad para la importación del hash de contraseñas

Amazon Cognito aplica medidas de seguridad adicionales a los hashes de contraseñas importados:

- Doble hash: Amazon Cognito vuelve a aplicar el hash de todas las contraseñas importadas con una capa adicional de protección criptográfica antes de almacenarlas, independientemente de la solidez del algoritmo original.
- Migración automática: tras una primera autenticación correcta, Amazon Cognito migra las credenciales del usuario al protocolo Secure Remote Password (SRP) nativo que utiliza Amazon Cognito. Esto garantiza que todos los usuarios utilicen finalmente el mecanismo de autenticación nativo de alta seguridad.
- Limpieza de datos: una vez finalizado el trabajo de importación, Amazon Cognito elimina del almacenamiento temporal el archivo CSV cargado que contiene los hashes de las contraseñas.
- Política de contraseñas no aplicada: Amazon Cognito no aplica la política de contraseñas que configuró para el grupo de usuarios al importar usuarios con hashes de contraseñas. Si una contraseña no cumple con la política de contraseñas que configuró, Amazon Cognito seguirá importando el usuario. Para reforzar la seguridad de las contraseñas y rechazar las contraseñas que no cumplan con la política, valide la seguridad de las contraseñas en el sistema de origen antes de generar hashes para la importación.

> **Important**
>
> Trata el archivo CSV que contiene los hashes de las contraseñas como datos confidenciales. Protéjalo en tránsito y elimínelo de forma segura una vez finalizada la importación.

Uso de atributos de usuario

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 Consola de administración de AWS. 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.

Desinfecte las entradas de los valores de cadena de atributos de usuario antes de enviarlos a su grupo de usuarios. Un método para analizar los valores de los atributos de usuario propuestos es emplear un desencadenador de Lambda, como Antes del registro.

> **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:

- `name`
- `family_name`
- `given_name`
- `middle_name`
- `nickname`
- `preferred_username`

- `profile`
- `picture`
- `website`
- `gender`
- `birthdate`
- `zoneinfo`
- `locale`
- `updated_at`
- `address`
- `email`
- `phone_number`
- `sub`

A excepción de `sub`, los atributos estándar son opcionales de forma predeterminada para todos los usuarios. Para hacer que un atributo sea obligatorio, durante el proceso de creación del grupo de usuarios, seleccione la casilla Obligatorio 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.

Los atributos estándar tienen propiedades predefinidas que se pueden ver en el `SchemaAttributes` parámetro de una respuesta de la DescribeUserPool API. Puede establecer valores personalizados para estas propiedades de atributo, como el tipo de datos, la mutabilidad o las restricciones de longitud. Para modificar las propiedades de los atributos estándar, defina sus valores personalizados en el parámetro CreateUserPool Schema. En este parámetro también debe configurar los atributos necesarios. Las propiedades de los atributos estándar no se pueden modificar al crear grupos de usuarios en la consola de Amazon Cognito.

> **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 en el formato YYYY-MM-DD.

correo electrónico

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 AWS Command Line Interface admin-update-user-attributes ().AWS CLI Este comando permite al administrador cambiar el atributo `email_verified` a `true`. También puede editar un usuario en el menú Usuarios de la consola de Amazon Cognito 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 comando admin-update-user-attributes AWS CLI . Este comando permite al administrador cambiar el atributo `phone_number_verified` a `true`. También puede editar un usuario en el menú Usuarios de la consola de Amazon Cognito 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, puedes realizar una solicitud a la operación de la UpdateUserAttributesAPI y añadir el valor del atributo después de confirmar al 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.

> **Important**
>
> Amazon Cognito genera `sub` en un formato de Amazon que no se ajusta a un Cognito-specific formato de UUID específico, incluido el UUID de RFC. No debe validar estrictamente el formato de. `sub`

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. Consola de administración de AWS 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. Seleccione el menú Sign-up.
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
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] Su grupo de usuarios no genera `UsernameExistsException` errores cuando los usuarios se registran con direcciones de correo electrónico o números de teléfono potencialmente duplicados, pero sin un 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)

Un atributo es un alias cuando los usuarios tienen un nombre de usuario, pero también pueden iniciar sesión con ese atributo. Cree los alias si quiere que los usuarios tengan la opción de elegir entre el nombre de usuario y otros valores de atributos en el campo de nombre de usuario del formulario de inicio de sesión. El atributo `username` 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 alias también se pueden transferir entre cuentas cuando un usuario verifica un atributo con. `VerifyUserAttribute` Si un usuario verifica una dirección de correo electrónico o un número de teléfono que ya es un alias verificado en otra cuenta, Amazon Cognito transfiere el alias a su cuenta y marca el atributo como no verificado en la cuenta original. Esta transferencia no genera un. `AliasExistsException`

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.

Phone number or email address (console)

Debe configurar la dirección de correo electrónico y el número de teléfono como atributos de alias al crear un grupo de usuarios.

Cómo crear un grupo de usuarios con los alias de los nombres de usuario en la consola de Amazon Cognito

1. Diríjase a Amazon Cognito en la Consola de administración de AWS. Si la consola se lo pide, introduce tus AWS credenciales.
2. Cree un nuevo grupo de usuarios con el botón Introducción o Crear grupo de usuarios.
3. Elija la configuración de la aplicación en Defina su aplicación.
4. En Configurar opciones, en Opciones para los identificadores de inicio de sesión, seleccione la casilla junto a Nombre de usuario y, al menos, una de las otras opciones, Correo electrónico y Número de teléfono.

5. Ponga los atributos de su alias como Atributos necesarios para el inicio de sesión. En el formulario de registro del inicio de sesión administrado, Amazon Cognito pide a los nuevos usuarios que proporcionen valores para los atributos obligatorios.
6. En Agregar una URL de retorno, configure una URL de devolución de llamada de la aplicación para redirigirla tras abrir sesión con el inicio de sesión administrado.
7. Seleccione Crear.

Phone number or email address (API/SDK)

Cree un nuevo grupo de usuarios con la operación de la CreateUserPoolAPI. Configure el parámetro `AliasAttributes` como se muestra. Puede eliminar la entrada `email` si solo quiere alias de números de teléfono o eliminar la entrada `phone_number` si solo quiere alias de direcciones de correo electrónico.

```
"AliasAttributes": [
   "email",
   "phone_number"
],
```

Preferred username (API/SDK)

La consola de Amazon Cognito crea grupos de usuarios sin `preferred_username` como alias. Para crear grupos de usuarios con un `preferred_username` alias, configure grupos de usuarios con solicitudes de CreateUserPoolAPI en un AWS SDK. Para permitir la creación de los atributos de nombre de usuario preferentes en el registro, configure `preferred_username` como atributo obligatorio. En el formulario de registro del inicio de sesión administrado, Amazon Cognito pide a los nuevos usuarios que proporcionen valores para los atributos obligatorios. Puede configurar `preferred_username` como atributo obligatorio en la consola de Amazon Cognito, pero esto no significa que esté disponible como alias.

Configuración como alias

Configure `preferred_username` como un alias en el parámetro `AliasAttributes` de una solicitud `CreateUserPool`, como se muestra a continuación. Elimine de la lista los valores que no desee utilizar como atributos de alias.

```
"AliasAttributes": [
   "email",
```

```
   "phone_number",
   "preferred_username"
],
```

Configuración como obligatorio

En el formulario de registro del inicio de sesión administrado, Amazon Cognito pide a los nuevos usuarios que proporcionen valores para los atributos obligatorios. `preferred_username`Configúrelo según sea necesario en el `SchemaAttributes` parámetro de una CreateUserPoolsolicitud.

Para establecer el nombre de usuario preferente como atributo obligatorio, configúrelo como se muestra a continuación. El siguiente ejemplo modifica el esquema predeterminado de `preferred_username` para configurarlo como obligatorio. Otros parámetros del esquema, como `AttributeDataType` (cuyo valor predeterminado es `string`) y `StringAttributeConstraints` (cuyo valor predeterminado tiene entre 1 y 99 caracteres de longitud) asumen valores predeterminados.

```
"Schema": [
   {
      "Name": "preferred_username",
      "Required": true
   }
]
```

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.

Al seleccionar tanto la dirección de correo electrónico como el número de teléfono como atributos de nombre de usuario, los usuarios pueden iniciar sesión con uno u otro, incluso si proporcionan valores para ambos atributos. El nombre de usuario de inicio de sesión se basa en el valor que se introduce en el `Username` parámetro. SignUp

> **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.

Username attributes (console)

El siguiente procedimiento crea un grupo de usuarios con dirección de correo electrónico o número de teléfono en los atributos de nombre de usuario. La diferencia en el proceso de los atributos de username en la consola de Amazon Cognito es que no se establece también el nombre de usuario como atributo de inicio de sesión.

Cómo crear un grupo de usuarios con atributos de nombres de usuario en la consola de Amazon Cognito

1. Diríjase a Amazon Cognito en la Consola de administración de AWS. Si la consola se lo pide, introduzca sus credenciales. AWS
2. Cree un nuevo grupo de usuarios con el botón Introducción o Crear grupo de usuarios.
3. Elija la configuración de la aplicación en Defina su aplicación.
4. En Configurar opciones, en Opciones para los identificadores de inicio de sesión, seleccione sus atributos de nombre de usuario: Correo electrónico, Número de teléfono o ambos. Deje Nombre de usuario sin marcar.

5. Como práctica recomendada, seleccione los atributos de su nombre de usuario como Atributos necesarios para el inicio de sesión. En el formulario de registro del inicio de sesión administrado, Amazon Cognito pide a los nuevos usuarios que proporcionen valores para los atributos obligatorios. Si no establece los atributos de su nombre de usuario como obligatorios, Amazon Cognito no solicitará a los nuevos usuarios que proporcionen valores para ellos. En ese escenario, debe configurar la aplicación para que recopile y envíe las direcciones de correo electrónico o los números de teléfono de cada usuario antes de que puedan iniciar sesión.
6. En Agregar una URL de retorno, configure una URL de devolución de llamada de la aplicación para redirigirla tras abrir sesión con el inicio de sesión administrado.
7. Seleccione Crear.

Username attributes (API/SDK)

En una CreateUserPoolsolicitud, configure el `UsernameAttributes` parámetro como se muestra. Para permitir el inicio de sesión solo con nombres de usuario de direcciones de correo electrónico, ponga solo `email` en esta lista. Para permitir el inicio de sesión únicamente con nombres de usuario de números de teléfono, ponga solo `phone_number`. Este parámetro anula el nombre de usuario como opción de inicio de sesión.

```
"UsernameAttributes": [
   "email",
   "phone_number"
],
```

El siguiente ejemplo configura un grupo de usuarios en el que cada usuario inicia sesión únicamente con su dirección de correo electrónico como nombre de usuario. Como práctica recomendada, establézcalo también `email` como atributo de registro obligatorio para que los nuevos usuarios proporcionen su dirección de correo electrónico cuando se registren mediante el inicio de sesión gestionado:

```
"UsernameAttributes": [
   "email"
],
```

Al configurar los atributos del nombre de usuario, puede realizar solicitudes de SignUpAPI que incluyan una dirección de correo electrónico o un número de teléfono en el `username` parámetro.

El siguiente es el comportamiento de la operación de la SignUp API con los atributos de nombre de usuario.

- Si la cadena username tiene un formato de correo electrónico válido (como user@example.com), 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 (como +12065551212), 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.
- 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.
- La API SignUp rellena el atributo username con un UUID para el usuario. Este UUID tiene el mismo valor que la notificación sub en el token de identidad de usuario.

Puedes usar una dirección de correo electrónico o un número de teléfono en lugar del nombre de usuario en todas las API, excepto en la ListUsersoperación. Esto incluye las operaciones posteriores al registro, como ConfirmSignUpy. ResendConfirmationCode Aunque SignUp establece el username atributo en el UUID, puedes pasar la dirección de correo electrónico o el número de teléfono del usuario como Username valor de la solicitud. SignUp En las solicitudes de API ListUsers, puede especificar un Filter de email o phone_number. Para filtrar porusername, introduce el nombre de usuario del UUID. No puedes filtrar por la dirección de correo electrónico o el número de teléfono de un username filtro.

Custom attributes (Atributos personalizados)

Puede añadir hasta 50 atributos personalizados a un grupo de usuarios. Puede especificar una longitud mínima, una longitud máxima o ambas para los atributos personalizados. Sin embargo, la longitud máxima de ningún atributo personalizado puede superar los 2048 caracteres. El nombre de un atributo personalizado debe coincidir con el patrón de expresión regular que se describe en el Name parámetro de SchemaAttributeType.

⚠ Los atributos personalizados son permanentes

No puede eliminar ni cambiar un atributo personalizado después de añadirlo a su grupo de usuarios. No puede eliminar ni cambiar el nombre del atributo. Tampoco puede cambiar sus

restricciones de tipo de datos, mutabilidad o longitud. Planifique su esquema de atributos personalizado antes de añadir atributos a un grupo de usuarios de producción.

Cada atributo personalizado incluye las siguientes características:

- Puede definirlo como cadena, número, booleano o un objeto `DateTime`. Amazon Cognito escribe valores de atributos personalizados en el token de ID solo como cadenas.

  > **Note**
  >
  > En la consola de Amazon Cognito, solo puede añadir atributos personalizados de los tipos de datos de cadena y número. Las opciones adicionales, como los tipos de datos booleanos y de `DateTime` atributos, solo están disponibles en las solicitudes de `SchemaAttributes` propiedad CreateUserPooly UpdateUserPoolAPI.

  > **Note**
  >
  > Independientemente del tipo de datos que asigne, Amazon Cognito almacena y transmite cada valor de atributo personalizado en forma de cadena. Cuando establece un valor en una solicitud de API o SDK (por ejemplo AdminUpdateUserAttributesUpdateUserAttributes, o SignUp), debe pasar el valor en forma de cadena, incluso para un `Number` `DateTime` atributo o un atributo. `Boolean` Pasar `"42"` en lugar de`42`, y `"true"` en lugar de`true`. Se rechaza un número nativo o un booleano antes de almacenar el valor; en función del SDK, se trata de un error de validación de tipo por parte del cliente o de un servicio. `InvalidParameterException` Tu aplicación es responsable de volver a analizar la cadena hasta el tipo deseado cuando lee el atributo.

- 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. Consola de administración de AWS 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. Seleccione el Sign-upmenú y, en la sección Atributos personalizados, elija 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. Seleccione Save changes (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 crea un nuevo cliente de aplicación, concede a la aplicación 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, especifique los permisos de lectura y escritura de los atributos al crear el cliente de aplicación. Concédale a este acceso al conjunto mínimo de atributos de usuario que son necesarios para que funcione la aplicación.

> **Important**
>
> Si no especificas un cliente `WriteAttributes` de aplicación, el cliente de aplicación tiene acceso de escritura a todos los atributos estándar y personalizados. Los usuarios pueden cambiar el valor de estos atributos por sí mismos con`UpdateUserAttributes`.
> La aplicación o el grupo de identidades pueden tomar decisiones de autorización en función de un atributo. En ese caso, elimine el atributo de todos los clientes `WriteAttributes` de aplicaciones de su grupo de usuarios. Todos los clientes de aplicaciones de un grupo de usuarios comparten los atributos de usuario. El valor que un usuario escribe a través de un cliente de aplicación aparece en los tokens que emiten todos los clientes de aplicaciones.

> **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 (Consola de administración de AWS)

1. Vaya a Amazon Cognito en. Consola de administración de AWS 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. Seleccione el menú Clientes de aplicación y seleccione un cliente de aplicación de la lista.
5. En la pestaña Permisos de atributos, elija 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. Para hacer que un atributo sea de solo lectura en sus aplicaciones, elimine el permiso de escritura para ese atributo de todos los clientes de aplicaciones de su grupo de usuarios.

Por cada cliente de 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 el mensaje. `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.
- Sign-in con un IdP de terceros: un usuario inicia sesión en un cliente de aplicaciones 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. Esto es posible, pero no es práctico, ya que el usuario solo podrá iniciar sesión una vez. En los intentos de inicio de sesión posteriores, Amazon Cognito lo rechazará debido a la regla de asignación a un atributo en el que ya no se podrá escribir.
- `AdminCreateUser`: usted proporciona un valor para un atributo inmutable.

Permisos de atributos con ámbitos

En los grupos de usuarios que configure con un AWS SDK o un CDK, la API REST o la AWS CLI, puede configurar el acceso de lectura o escritura del cliente de aplicaciones con el ámbito del OIDC. `oidc:profile` `oidc:profile` otorga acceso de lectura o escritura a los siguientes atributos estándar:

- `name`
- `family_name`
- `given_name`
- `middle_name`
- `nickname`
- `preferred_username`
- `profile`
- `picture`
- `website`
- `gender`
- `birthdate`
- `zoneinfo`
- `locale`

Esta lista contiene los atributos estándar de OIDC menos `email`, `phone_number`, `sub` y `address`, según la definición que figura en la sección 2.4 de la especificación de OIDC. Para obtener

información sobre los ámbitos que puede asignar a los clientes de la aplicación, consulte Ámbitos, M2M y servidores de recursos.

Para configurar el cliente de la aplicación para que escriba en los atributos incluidos en el `oidc:profile` ámbito, defina el valor de WriteAttributestoo`idc:profile`, además de cualquier otro atributo que desee permitir que la aplicación modifique, en una solicitud CreateUserPoolCliento UpdateUserPoolClienta la API. Del mismo modo, para conceder acceso de lectura a estos atributos, añade `oidc:profile` el valor de ReadAttributes.

Puede cambiar los permisos y los alcances de los atributos después de crear el grupo de usuarios.

Actualización de la configuración del grupo de usuarios y del cliente de aplicación

Cuando desee cambiar un ajuste de un grupo de usuarios o de un cliente de aplicación, puede aplicar la actualización en la consola de Amazon Cognito con solo unos cuantos clics. Desplácese por las pestañas de características de la configuración del grupo de usuarios y actualice los campos tal como se describe en otras secciones de esta guía.

Muchas organizaciones administran sus recursos mediante programación en aplicaciones AWS CloudFormation basadas en los AWS SDK o CDK y otro software de automatización. Si este es su modelo de administración de recursos, debe tener especial cuidado al introducir cambios en los recursos.

La API funciona UpdateUserPooly UpdateUserPoolClientactualiza un grupo de usuarios o un cliente de aplicaciones existentes. Cada una incluye una advertencia en la referencia de la API: Si no proporciona un valor a un atributo, Amazon Cognito lo establecerá en su valor predeterminado. Cuando envía una solicitud de actualización con un único parámetro, Amazon Cognito establece ese parámetro en el valor que ha elegido y establece todos los demás en el valor predeterminado. Esto puede restablecer las configuraciones, incluido el esquema de atributos, los desencadenadores de Lambda y la configuración del correo electrónico y los mensajes SMS.

Además, algunos ajustes se bloquean después de crear el grupo de usuarios o el cliente de aplicación y no se pueden cambiar a menos que cree un recurso nuevo.

Temas

- Ajustes que no se pueden cambiar

- Configuración de SMS
- Actualización de un grupo de usuarios con un AWS SDK, AWS CDK, o API REST

Ajustes que no se pueden cambiar

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.

> **Note**
>
> Antes, no se podía cambiar el nombre de un grupo de usuarios. Esto ha cambiado. Ahora puede asignar nuevos nombres descriptivos a sus grupos de usuarios.

ID de grupo de usuarios

Nombre del parámetro de la API: Id/UserPoolId

Amazon Cognito genera automáticamente el ID del grupo de usuarios, como `us-east-1_EXAMPLE`, y no se puede cambiar.

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 al iniciar sesión. Al crear un grupo de usuarios, puede optar por permitir el inicio de sesión con el nombre de usuario, la dirección de correo electrónico, el número de teléfono o el nombre de usuario que prefiera. 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 crea un nombre de usuario que coincide con otro nombre de usuario, excepto las letras mayúsculas y minúsculas, Amazon Cognito puede tratarlos 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 Application-specific configuración con clientes de aplicaciones. Para cambiar un secreto de cliente, cree un nuevo cliente de aplicaciones en el mismo grupo de usuarios.

Atributos obligatorios.

Nombre del parámetro de la API: Schema

Los atributos para los que los usuarios deben proporcionar valores cuando se registran o cuando los crea. Para obtener más información, consulte Uso de atributos de usuario. Para cambiar los atributos obligatorios, cree un nuevo grupo de usuarios.

Atributos personalizados (eliminación)

Nombre del parámetro de la 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 Uso de atributos de usuario. Si alcanza el número máximo de atributos personalizados y desea modificar la lista, cree un nuevo grupo de usuarios.

Configuración de SMS

Una vez que haya activado los mensajes SMS en el grupo de usuarios, no los podrá desactivar.

- Si elige configurar mensajes SMS al crear un grupo de usuarios, no podrá desactivarlos una vez que haya completado la configuración.
- Puede activar los mensajes SMS en un grupo de usuarios que haya creado, pero después no podrá desactivarlos.
- Amazon Cognito puede utilizar los mensajes SMS para la invitación y la recuperación de cuentas de usuario, la verificación de atributos y la autenticación multifactor (MFA). Tras activar los mensajes SMS, puede activarlos o desactivarlos para estas funciones en cualquier momento.
- La configuración de mensajes SMS incluye un rol de IAM que puede delegar en Amazon Cognito para enviar mensajes con Amazon SNS. Puede cambiar el rol asignado en cualquier momento.

Actualización de un grupo de usuarios con un AWS SDK, AWS CDK, o API REST

En la consola de Amazon Cognito, puede cambiar la configuración del grupo de usuarios, parámetro por parámetro. Por ejemplo, para agregar un desencadenador de Lambda, elija primero Agregar desencadenador de Lambda y luego la función y el tipo de desencadenador. 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 aplicación requieran el conjunto completo de parámetros del grupo de usuarios. Sin embargo, la consola automatiza de forma transparente esta operación de actualización con los demás ajustes del grupo de usuarios.

Es posible que en ocasiones descubras que un cambio en alguna parte de la página Cuenta de AWS puede provocar que las actualizaciones generen un error cuando no están relacionadas con la configuración que deseas 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 se encuentre con un error de este tipo, examine la respuesta al error y valide la configuración que menciona.

Kit de desarrollo de la nube de AWS (AWS CDK), la API de REST de los grupos de usuarios de Amazon Cognito y los AWS SDK son herramientas para automatizar y configurar mediante programación los recursos de Amazon Cognito. Al igual que la consola de Amazon Cognito, las solicitudes con estas herramientas también deben actualizar un ajuste con una configuración de recursos completa en el cuerpo de la solicitud. En líneas generales, 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 el 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
```

Personalización de grupos de usuarios

Los grupos de usuarios de Cognito exponen los puntos de extensión para que pueda configurar la experiencia de registro e inicio de sesión de su aplicación. Utilice los activadores Lambda para ejecutar su propio código en momentos clave del ciclo de vida del usuario y configure los mensajes y los análisis que envía su grupo de usuarios.

Temas

- Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda
- Mensajería y análisis

Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda

Amazon Cognito trabaja con AWS Lambda funciones 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 del flujo de autenticación, realizar solicitudes de API para modificar el grupo de usuarios u otros AWS recursos 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. Un evento devuelto sin modificar notifica a su grupo de usuarios que debe continuar con la acción predeterminada para el evento. Por ejemplo, el desencadenador Antes del registro puede confirmar automáticamente a los usuarios para el origen del desencadenador `PreSignUp_SignUp`, pero devolver el evento sin cambios en el caso de los usuarios externos y creados por el administrador.

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	Description (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 “Antes de la autenticación”	Validación personalizada para aceptar o denegar la solicitud de inicio de sesión
	the section called “Después de la autenticación”	Registra eventos para los análisis personalizados
	the section called “Antes de la generación del token”	Aumenta o suprime las notificaciones de tokens
Federación	the section called “Federación entrante”	Transforma los atributos de los usuarios federados antes

Flujo del grupo de usuarios	Operación	Description (Descripción)
		de crearlos o actualizarlos en los grupos de usuarios de Amazon Cognito
Sign-Up	the section called “Antes del registro”	Realiza una validación personalizada que acepta o rechaza la solicitud de inscripción
	the section called “Después de la confirmación”	Agrega mensajes de bienvenida personalizados o el registro de eventos para los análisis personalizados
	the section called “Migración de usuario”	Migra un usuario desde un directorio de usuarios existente a los grupos de usuarios
Mensajes	the section called “Mensaje personalizado”	Realiza una personalización avanzada y localiza mensajes
Creación de tokens	the section called “Antes de la generación del token”	Añade o elimina atributos en tokens de identificación
Proveedores externos de correo electrónico y SMS	the section called “Remitentes personalizados”	Usa un proveedor de terceros para enviar mensajes SMS y de correo electrónico

Temas

- Conceptos importantes sobre los desencadenadores de Lambda
- 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
- Metadatos del cliente

- 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 Antes del registro.
- Desencadenador de Lambda de posconfirmación.
- Desencadenador de Lambda Antes de la autenticación
- Desencadenador de Lambda Después de la autenticación
- Activador Lambda de federación entrante
- 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

Conceptos importantes sobre los desencadenadores de Lambda

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 una Desencadenador de Lambda anterior a la generación del token, debe configurar su grupo de usuarios con un plan de características que no sea Lite y actualizar la configuración del desencadenador de Lambda para utilizar la versión 2 de eventos.
- 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 podría volver a intentarlo. Si todos los reintentos fallan, se

agota el tiempo de espera de la función. No puede cambiar ese valor de tiempo de espera de cinco segundos. Para obtener más información, consulte el modelo de programación Lambda en la Guía para AWS Lambda desarrolladores.

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 Gestión 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. Sin embargo, puedes asociar una versión de función `LambdaArn` a un alias y configurar tu activador en el alias ARN en una CreateUserPoolsolicitud de UpdateUserPoolAPI. Esta opción no está disponible en la Consola de administración de AWS. 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.

  El inicio de sesión administrado 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. Usa métodos de salida, como `print()` registrar cualquier información confidencial o de depuración en los CloudWatch registros. Para ver un ejemplo, consulta Ejemplo previo al registro: deniega el registro si el nombre de usuario tiene menos de cinco caracteres.
- Puede añadir una función Lambda en otra Cuenta de AWS como activador para su grupo de usuarios. Debe añadir activadores multicuenta con las operaciones de UpdateUserPoolAPI CreateUserPooly las operaciones de API, o sus equivalentes en y en CloudFormation . AWS CLI No puedes añadir funciones multicuenta en. Consola de administración de AWS

- 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 añadir permisos desde la consola de Lambda o utilizar la operación de la API de AddPermissionLambda.

  Ejemplo de política Lambda Resource-Based

  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`.

  JSON

```
{
    "Version":"2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "LambdaCognitoIdpTrust",
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-idp.amazonaws.com"
            },
            "Action": "lambda:InvokeFunction",
            "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:MyFunction",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "AWS:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        }
    ]
```

```
}
```

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. Seleccione el menú Extensiones y localice los desencadenadores de 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 Asignar función Lambda y seleccione una función Región de AWS igual a la de su grupo de usuarios.

   > **Note**
   >
   > Si sus credenciales AWS Identity and Access Management (de IAM) tienen permiso para actualizar la función Lambda, Amazon Cognito añade una política de Lambda basada en recursos. 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 “Cosas que debe saber”.

8. Elija Save changes (Guardar cambios).
9. Puede utilizarla CloudWatch en la consola Lambda para registrar la función Lambda. Para obtener más información, consulte Acceso a CloudWatch los registros de 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. Si la función devuelve el evento de entrada sin modificarlo, Amazon Cognito procederá con el comportamiento predeterminado. A continuación, se muestran los parámetros que son comunes a todos los eventos de entrada del desencadenador de Lambda. Para conocer la sintaxis de los eventos específicos de cada desencadenador, consulte el esquema de eventos de la sección de esta guía para cada desencadenador.

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).

region

El Región de AWS como `AWSRegion` instancia.

usuario PoolId

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 aws SdkVersion y ClientiD.

AWS SdkVersion

La versión del AWS SDK que generó la solicitud.

clientId

El ID de cliente de la aplicación del grupo de usuarios.

solicitud

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 activador de autenticación previa también contiene 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"`.

response

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.

Metadatos del cliente

Puede enviar parámetros personalizados a sus funciones de desencadenación de Lambda en las operaciones API y las solicitudes Punto de conexión de token. Con los metadatos del cliente, su aplicación puede recopilar información adicional sobre el entorno en el que se originan las solicitudes. Al pasar los metadatos del cliente a las funciones de Lambda, estas pueden procesar los datos adicionales y utilizarlos para registrar o personalizar los flujos de autenticación. Los metadatos del cliente son pares de cadenas que usted elige y diseña en formato clave-valor JSON.

Casos de uso de ejemplo de metadatos de cliente

- Transfiera los datos de geolocalización en el momento del registro al desencadenador Antes del registro y evite el inicio de sesión desde ubicaciones no deseadas.
- Transfiera los datos de identificación del inquilino a desencadenadores de desafíos personalizados y presente diferentes desafíos a los clientes de distintas unidades de negocio.
- Pase el token de un usuario al desencadenador Antes de la generación del token y genere un registro de la entidad principal en nombre del cual se realizó una solicitud M2M. Para obtener una solicitud de ejemplo, consulte Credenciales de cliente con autorización básica.

Este es un ejemplo de cómo pasar los metadatos del cliente al desencadenador Antes del registro.

SignUp request

El siguiente es un ejemplo de SignUpsolicitud con metadatos de cliente que Amazon Cognito transfiere a un activador previo al registro.

```
POST HTTP/1.1
Host: cognito-idp.us-east-1.amazonaws.com
X-Amz-Date: 20230613T200059Z
Accept-Encoding: gzip, deflate, br
X-Amz-Target: AWSCognitoIdentityProviderService.SignUp
User-Agent: <UserAgentString>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>

{
    "ClientId": "1example23456789",
    "Username": "mary_major",
    "Password": "<Password>",
```

```
    "SecretHash": "<Secret hash>",
    "ClientMetadata": {
        "IpAddress" : "192.0.2.252",
        "GeoLocation" : "Netherlands (Kingdom of the) [NL]"
    }
    "UserAttributes": [
        {
            "Name": "name",
            "Value": "Mary"
        },
        {
            "Name": "email",
            "Value": "mary_major@example.com"
        },
        {
            "Name": "phone_number",
            "Value": "+12065551212"
        }
    ],
}
```

Lambda trigger input event

La solicitud da como resultado el siguiente cuerpo de solicitud para su función Antes del registro.

```
{
    "callerContext": {
        "awsSdkVersion": "aws-sdk-unknown-unknown",
        "clientId": "1example23456789"
    },
    "region": "us-west-2",
    "request": {
        "clientMetadata": {
            "GeoLocation": "Netherlands (Kingdom of the) [NL]",
            "IpAddress": "192.0.2.252"
        },
        "userAttributes": {
            "email": "mary_major@example.com",
            "name": "Mary",
            "phone_number": "+12065551212"
        },
        "validationData": null
    },
    "response": {
```

```
        "autoConfirmUser": false,
        "autoVerifyEmail": false,
        "autoVerifyPhone": false
    },
    "triggerSource": "PreSignUp_SignUp",
    "userName": "mary_major2",
    "userPoolId": "us-west-2_EXAMPLE",
    "version": "1"
}
```

Metadatos del cliente para credenciales de cliente máquina a máquina (M2M)

Puede pasar los metadatos del cliente en las solicitudes M2M. Los metadatos del cliente son información adicional de un entorno de usuario o aplicación que puede contribuir a los resultados de una Desencadenador de Lambda anterior a la generación del token. En las operaciones de autenticación con un usuario principal, puede pasar los metadatos del cliente al activador previo a la generación del token, que se encuentra en el cuerpo de AdminRespondToAuthChallengelas solicitudes a la RespondToAuthChallengeAPI. Dado que las aplicaciones dirigen el flujo de generación de tokens de acceso para M2M con solicitudes directas al Punto de conexión de token, tienen un modelo diferente. En el cuerpo POST de las solicitudes de token de credenciales de cliente, pasa un `aws_client_metadata` parámetro con el objeto de metadatos del cliente URL-encoded (`x-www-form-urlencoded`) a la cadena. Para obtener una solicitud de ejemplo, consulte Credenciales de cliente con autorización básica. A continuación, puede ver un ejemplo de parámetro que transfiere los pares clave-valor `{"environment": "dev", "language": "en-US"}`.

```
aws_client_metadata=%7B%22environment%22%3A%20%22dev%22,%20%22language%22%3A%20%22en-US
%22%7D
```

Atributos de usuario temporales: **`ValidationData`**

Algunas operaciones de autenticación también tienen un parámetro `validationData`. Al igual que los metadatos de los clientes, esta es una oportunidad para pasar información externa que Amazon Cognito no recopila automáticamente a los desencadenadores de Lambda. El campo de datos de validación está diseñado para proporcionar a la función Lambda un contexto de usuario adicional en las operaciones de registro e inicio de sesión. SignUpy AdminCreateUserpasa `validationData` al activador previo al registro. InitiateAuthy AdminInitiateAuthpasa el cuerpo `ClientMetadata` de la solicitud de la API como si `validationData` fuera el evento de entrada a los activadores de preautenticación y migración de los usuarios.

Para asignar las operaciones de la API a las funciones a las que pueden pasar los metadatos del cliente, consulte las secciones sobre los orígenes de activación que aparecen a continuación.

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, el inicio de sesión administrado 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 Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado. 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
- Desencadenadores de Lambda para los usuarios locales de Amazon Cognito en el inicio de sesión administrado
- 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	Antes del registro	`PreSignUp_AdminCreateUser`
	Antes de la generación del token	`TokenGeneration_NewPasswordChallenge`

Operación de la API	Disparador de Lambda	Origen del disparador
	Mensaje personalizado	`CustomMessage_AdminCreateUser`
	Remitente de correo electrónico personalizado	`CustomEmailSender_AdminCreateUser`
	Remitente de SMS personalizado	`CustomSMSSender_AdminCreateUser`
SignUp	Antes del registro	`PreSignUp_SignUp`
	Mensaje personalizado	`CustomMessage_SignUp`
	Remitente de correo electrónico personalizado	`CustomEmailSender_SignUp`
	Remitente de SMS personalizado	`CustomSMSSender_SignUp`
ConfirmSignUp AdminConfirmSignUp	Después de la confirmación	`PostConfirmation_ConfirmSignUp`
InitiateAuth AdminInitiateAuth	Antes de la autenticación	`PreAuthentication_Authentication`
	Después de la autenticación	`PostAuthentication_Authentication`
	Definir desafío de autenticación	`DefineAuthChallenge_Authentication`
	Crear desafío de autenticación	`CreateAuthChallenge_Authentication`
	Verificación de desafío de autenticación	`VerifyAuthChallenge_Authentication`

Operación de la API	Disparador de Lambda	Origen del disparador

	Antes de la generación del token	`TokenGeneration_Authentication`
		`TokenGeneration_AuthenticateDevice`
		`TokenGeneration_RefreshTokens`

	Migración de usuario	`UserMigration_Authentication`
	Mensaje personalizado	`CustomMessage_Authentication`
	Remitente de correo electrónico personalizado	`CustomEmailSender_AccountTakeOverNotification` `CustomEmailSender_Authentication`
	Remitente de SMS personalizado	`CustomSMSSender_Authentication`
RespondToAuthChallenge AdminRespondToAuthChallenge	Después de la autenticación	`PostAuthentication_Authentication`
	Definir desafío de autenticación	`DefineAuthChallenge_Authentication`
	Crear desafío de autenticación	`CreateAuthChallenge_Authentication`
	Verificación de desafío de autenticación	`VerifyAuthChallenge_Authentication`

Operación de la API	Disparador de Lambda	Origen del disparador

	Antes de la generación del token	`TokenGeneration_Authentication`
		`TokenGeneration_AuthenticateDevice`
		`TokenGeneration_RefreshTokens`

	Mensaje personalizado	`CustomMessage_Authentication`
	Remitente de correo electrónico personalizado	`CustomEmailSender_AccountTakeOverNotification` `CustomEmailSender_Authentication`
	Remitente de SMS personalizado	`CustomSMSSender_Authentication`
ForgotPassword	Migración de usuario	`UserMigration_ForgotPassword`
	Mensaje personalizado	`CustomMessage_ForgotPassword`
	Remitente de correo electrónico personalizado	`CustomEmailSender_ForgotPassword`
	Remitente de SMS personalizado	`CustomSMSSender_ForgotPassword`
ConfirmForgotPassword	Después de la confirmación	`PostConfirmation_ConfirmForgotPassword`

Operación de la API	Disparador de Lambda	Origen del disparador
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`
GetTokensFromRefreshToken	Antes de la generación del token	`TokenGeneration_Authentication`

Desencadenadores de Lambda para los usuarios locales de Amazon Cognito en el inicio de sesión administrado

En la siguiente tabla, se describen las cadenas de origen de los desencadenadores de Lambda que Amazon Cognito puede invocar cuando un usuario local inicia sesión en el grupo de usuarios con el inicio de sesión administrado.

Orígenes de desencadenadores de usuarios locales en el inicio de sesión administrado

URI en el inicio de sesión administrado	Disparador de Lambda	Origen del disparador
`/signup`	Antes del registro	`PreSignUp_SignUp`
	Mensaje personalizado	`CustomMessage_SignUp`

URI en el inicio de sesión administrado	Disparador de Lambda	Origen del disparador
	Remitente de correo electrónico personalizado	`CustomEmailSender_SignUp`
	Remitente de SMS personalizado	`CustomSMSSender_SignUp`
`/confirmuser`	Después de la confirmación	`PostConfirmation_ConfirmSignUp`
`/login`	Antes de la autenticación	`PreAuthentication_Authentication`

	Antes de la generación del token	`TokenGeneration_Authentication`
		`TokenGeneration_AuthenticateDevice`
		`TokenGeneration_RefreshTokens`

	Migración de usuario	`UserMigration_Authentication`
	Mensaje personalizado	`CustomMessage_Authentication`
	Remitente de correo electrónico personalizado	`CustomEmailSender_AccountTakeOverNotification` `CustomEmailSender_Authentication`
	Remitente de SMS personalizado	`CustomSMSSender_Authentication`

URI en el inicio de sesión administrado	Disparador de Lambda	Origen del disparador
`/forgotpassword`	Migración de usuario	`UserMigration_ForgotPassword`
	Mensaje personalizado	`CustomMessage_ForgotPassword`
	Remitente de correo electrónico personalizado	`CustomEmailSender_ForgotPassword`
	Remitente de SMS personalizado	`CustomSMSSender_ForgotPassword`
`/confirmforgotpassword`	Después de 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 el inicio de sesión administrado 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

Sign-in evento	Disparador de Lambda	Origen del disparador
Primer inicio de sesión	Antes del registro	`PreSignUp_ExternalProvider`

Sign-in evento	Disparador de Lambda	Origen del disparador
	Después de la confirmación	`PostConfirmation_ConfirmSignUp`
	Antes de la generación del token	`TokenGeneration_HostedAuth`
Inicios de sesión posteriores	Antes de la autenticación	`PreAuthentication_Authentication`
	Después de la autenticación	`PostAuthentication_Authentication`
	Antes de 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.

Sign-up, activadores de confirmación e inicio de sesión (autenticación)

Desencadenador	Valor de triggerSource	Event
Antes del registro	`PreSignUp_SignUp`	Antes del registro.
Antes del registro	`PreSignUp_AdminCreateUser`	Antes del registro cuando un administrador crea un nuevo usuario.
Antes del registro	`PreSignUp_ExternalProvider`	Antes del registro para proveedores de identidad externos.
Después de la confirmación	`PostConfirmation_ConfirmSignUp`	Posterior a la confirmación de la inscripción.
Después de la confirmación	`PostConfirmation_ConfirmForgotPassword`	Posterior a la confirmación de la contraseña olvidada.
Antes de la autenticación	`PreAuthentication_Authentication`	Antes de la autenticación.
Después de la autenticación	`PostAuthentication_Authentication`	Después de la autenticación.

Disparadores de desafío de autenticación personalizados

Desencadenador	Valor de triggerSource	Event
Definir desafío de autenticación	`DefineAuthChallenge_Authentication`	Definición de desafíos de autenticación.
Crear desafío de autenticación	`CreateAuthChallenge_Authentication`	Creación de desafíos de autenticación.

Desencadenador	Valor de triggerSource	Event
Verificación de desafío de autenticación	`VerifyAuthChallengeResponse_Authentication`	Verificación de la respuesta a los desafíos de autenticación.

Activadores de federación

Desencadenador	Valor de triggerSource	Event
Federación entrante	`InboundFederation_ExternalProvider`	Federación entrante.

Disparadores anteriores a la generación del token

Desencadenador	Valor de triggerSource	Event
Antes de la generación del token	`TokenGeneration_HostedAuth`	Amazon Cognito autentica al usuario desde su página de inicio de sesión en el inicio de sesión administrado.
Antes de la generación del token	`TokenGeneration_Authentication`	Se completó la autenticación del usuario o la actualización del token.
Antes de la generación del token	`TokenGeneration_NewPasswordChallenge`	El administrador crea el usuario. Amazon Cognito lo llama cuando el usuario debe cambiar una contraseña temporal.
Antes de la generación del token	`TokenGeneration_AuthenticateDevice`	Fin de la autenticación de un dispositivo de usuario.

Desencadenador	Valor de triggerSource	Event
Antes de 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	Event
Migración de usuario	`UserMigration_Authentication`	Migración de usuarios durante el inicio de sesión.
Migración de usuario	`UserMigration_ForgotPassword`	Migración de usuarios durante el flujo de recuperación de contraseñas olvidadas.

Disparadores de mensaje personalizado

Desencadenador	Valor de triggerSource	Event
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

Desencadenador	Valor de triggerSource	Event
		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.
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.

Desencadenadores de remitente personalizados

Desencadenador	Valor de triggerSource	Event

Remitente personalizado	`CustomEmailSender_SignUp`
	
	`CustomSmsSender_SignUp`
		Cuando un usuario se registra en su grupo de usuarios.

Remitente personalizado	`CustomEmailSender_AdminCreateUser`	Cuando crea un usuario como administrador y Amazon Cognito le envía una contraseña temporal.

Desencadenador	Valor de triggerSource	Event
	`CustomSmsSender_AdminCreateUser`	

Remitente personalizado	`CustomEmailSender_ForgotPassword`
	
	`CustomSmsSender_ForgotPassword`
		Cuando el usuario solicita un restablecimiento de contraseña.

Remitente personalizado	`CustomEmailSender_UpdateUserAttribute`
	
	`CustomSmsSender_UpdateUserAttribute`
		Cuando un usuario cambia su dirección de correo electrónico o número de teléfono y Amazon Cognito envía un código de verificación.

Remitente personalizado	`CustomEmailSender_VerifyUserAttribute`
	
	`CustomSmsSender_VerifyUserAttribute`
		Cuando un usuario agrega una dirección de correo electrónico o un número de teléfono y Amazon Cognito envía un código de verificación.

Remitente personalizado	`CustomEmailSender_Authentication`
	
	`CustomSmsSender_Authentication`
		Cuando un usuario que ha configurado la MFA u OTP por SMS o correo electrónico inicia sesión.

Remitente personalizado	`CustomEmailSender_AccountTakeOverNotification`	Cuando la configuración de protección contra amenazas adopta una acción automática contra el intento de inicio de sesión de un usuario y la acción correspondiente al nivel de riesgo incluye una notificación.

Desencadenador de Lambda Antes del registro.

Puede que quiera personalizar el proceso de registro en los grupos de usuarios que tienen opciones de registro de autoservicio. Normalmente, el desencadenador Antes del registro realiza análisis y registros personalizados de los nuevos usuarios, aplica estándares de seguridad y gobernanza o vincula a los usuarios de un IdP de terceros a un perfil de usuario consolidado. También es posible que tenga usuarios de confianza que no estén obligados a someterse a una verificación o confirmación.

Justo antes de completar la creación de un usuario local o federado nuevo, Amazon Cognito activa la función de Lambda anterior al registro. El `userAttributes` en el objeto de solicitud enviado a esta función contiene atributos que se han proporcionado al registrarse un usuario local o que se han asignado correctamente a partir de los atributos del proveedor para un usuario federado. Su grupo de usuarios invoca este activador al registrarse en el autoservicio SignUpo al iniciar sesión por primera vez con un proveedor de identidad de confianza y al crear usuarios con. AdminCreateUser Durante el proceso de registro puede utilizar esta función para analizar el evento de inicio de sesión con una lógica personalizada y modificar el usuario nuevo o rechazarlo.

Temas

- Parámetros del desencadenador de Lambda de prerregistro
- Ejemplo previo al registro: Auto-confirm usuarios de un dominio registrado
- Ejemplo previo al registro: Auto-confirm y verificación automática de todos los usuarios
- Ejemplo previo al registro: deniega el registro si el nombre de usuario tiene menos de cinco caracteres

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.

auto ConfirmUser

Establezca este parámetro en `true` para confirmar automáticamente al usuario, o en `false` en caso contrario.

auto VerifyEmail

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 `email` atributo se selecciona como alias, se crea un alias para la dirección de correo electrónico del usuario cuando `autoVerifyEmail` se establece. Si ya existe un alias con esa dirección de correo electrónico, el alias se mueve al nuevo usuario y la dirección de correo electrónico del usuario anterior se marca como no verificada. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

auto VerifyPhone

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 `phone_number` atributo se selecciona como alias, se crea un alias para el número de teléfono del usuario cuando `autoVerifyPhone` se establece. Si ya existe un alias con ese

número de teléfono, el alias se transfiere al nuevo usuario y el número de teléfono del usuario anterior se marca como no verificado. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

Ejemplo previo al registro: Auto-confirm usuarios de un dominio registrado

Este es un ejemplo de código de desencadenador de Lambda. El desencadenador Antes del registro se invoca inmediatamente antes de que Amazon Cognito procese la solicitud de registro. 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 estén en el dominio personalizado se añaden al grupo de usuarios, pero no se confirman automáticamente.

Node.js

```
export const handler = async (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 previo al registro: Auto-confirm y verificación automática 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 la creación de alias está habilitada, se crean alias para `phone_number` y `email` cuando se establece la verificación automática.

> **Note**
>
> Si ya existe un alias con el mismo número de teléfono, el alias se transfiere al nuevo usuario y el del usuario anterior `phone_number` se marca 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

```
exports.handler = (event, context, callback) => {
  // 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 to Amazon Cognito
  callback(null, event);
};
```

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 previo al registro: deniega el registro si el nombre de usuario tiene menos de cinco caracteres

En este ejemplo, se comprueba la longitud del nombre de usuario en 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

```
export const 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. Su grupo de usuarios invoca su función de confirmación de publicación en ConfirmSignUpAdminConfirmSignUp, y ConfirmForgotPassword. Este desencadenador también se ejecuta cuando los usuarios confirman el registro o el restablecimiento de la contraseña en un inicio de sesión administrado.

Temas

- Parámetros del desencadenador de Lambda de posconfirmación
- Ejemplo de invocación posterior a la confirmación

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 pasar estos datos a la función Lambda mediante el ClientMetadata parámetro de las siguientes acciones de la API: AdminConfirmSignUp, ConfirmForgotPasswordConfirmSignUp, y. SignUp

Parámetros de la respuesta posterior a la confirmación

No se espera que la respuesta contenga información adicional.

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 Antes de la autenticación

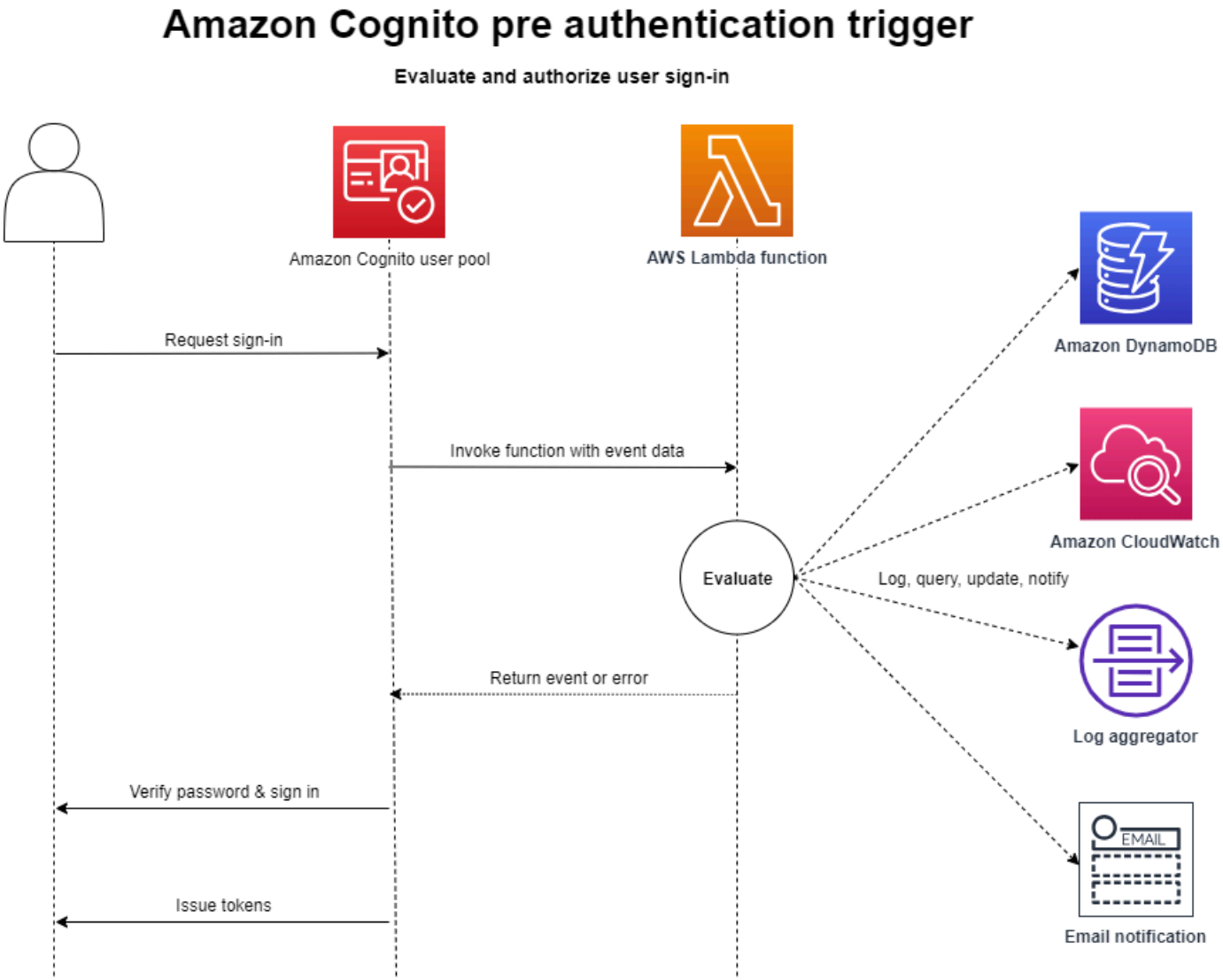
Amazon Cognito invoca a este desencadenador cuando un usuario intenta iniciar sesión, lo que le permite crear una validación personalizada que realiza acciones preparatorias. Por ejemplo, puede denegar la solicitud de autenticación o registrar los datos de sesión en un sistema externo.

> **Note**
>
> Este desencadenador de Lambda no se activa cuando un usuario no existe, a menos que el ajuste `PreventUserExistenceErrors` de un cliente de aplicación en un grupo de usuarios se haya establecido en `ENABLED`. La renovación de una sesión de autenticación existente tampoco activa este desencadenador.

Temas

- Información general sobre flujos
- Parámetros del desencadenador de Lambda de preautenticación
- Ejemplo invocación Antes de la autenticación

Información general sobre flujos



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 Un ejemplo de sesión de autenticación.

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 Antes de la autenticación

userAttributes

Uno o varios pares de nombre-valor que representan atributos de usuario.

usuario NotFound

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. Para pasar estos datos a la función Lambda, utilice el ClientMetadata parámetro en las acciones InitiateAuthy de la AdminInitiateAuthAPI.

Parámetros de la respuesta Antes de la autenticación

Amazon Cognito no procesa ninguna información adicional que su función devuelva 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 Antes de 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 Después de la autenticación

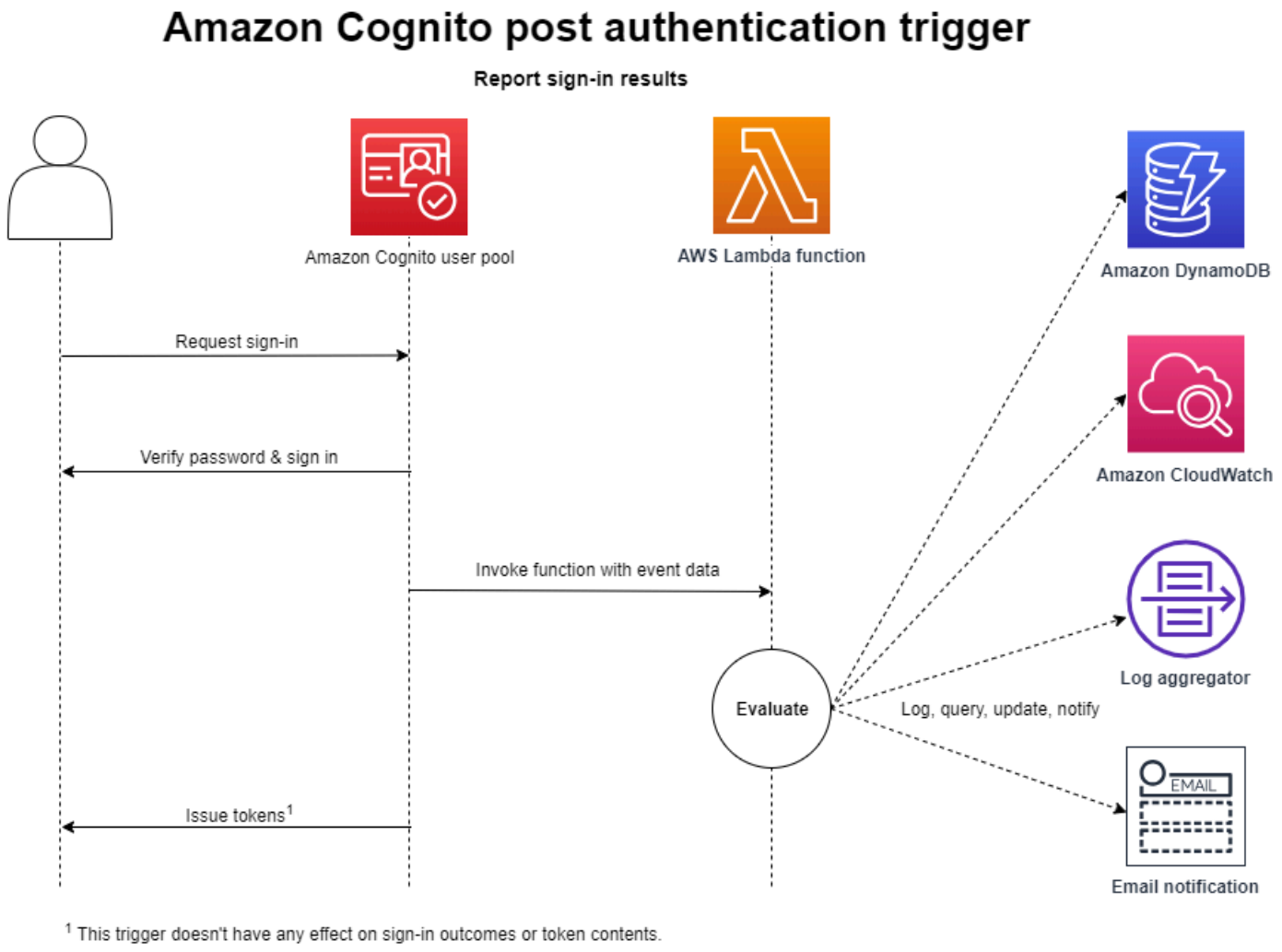
El desencadenador posterior a la autenticación no cambia el flujo de autenticación del usuario. Amazon Cognito invoca este desencadenador Lambda al finalizar la autenticación y antes de que el usuario reciba los tokens. Añada un activador posterior a la autenticación cuando desee añadir un posprocesamiento personalizado de los eventos de autenticación, por ejemplo, ajustes de registro o perfil de usuario que se reflejan en el siguiente inicio de sesión.

Un desencadenador de Lambda posterior a la autenticación que no devuelva el cuerpo de la solicitud a Amazon Cognito puede seguir siendo la causa de que la autenticación no consiga realizarse. Para obtener más información, consulte Conceptos importantes sobre los desencadenadores de Lambda.

Temas

- Información general sobre el flujo de autenticación
- Parámetros del desencadenador de Lambda de posautenticación
- Ejemplo de invocación Después de la autenticación

Información general sobre el flujo de autenticación



Para obtener más información, consulte Un ejemplo de sesión de autenticación.

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 Después de la autenticación

nuevo DeviceUsed

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. 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

Parámetros de la respuesta Después de 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.

Ejemplo de invocación Después de la autenticación

Este ejemplo de función Lambda posterior a la autenticación envía los datos de un inicio de sesión correcto a Logs. CloudWatch

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": {}
}
```

Activador Lambda de federación entrante

El activador de federación entrante transforma los atributos de los usuarios federados durante el proceso de autenticación con proveedores de identidad externos. Cuando los usuarios se autentican a través de proveedores de identidad configurados, este desencadenante le permite modificar las respuestas de proveedores de SAML y OIDC externos al interceptar y transformar los datos en el proceso de autenticación, lo que proporciona un control programático sobre la forma en que los grupos de usuarios de Amazon Cognito gestionan los usuarios federados y sus atributos.

Utilice este activador para añadir, anular o suprimir atributos antes de crear nuevos usuarios o actualizar los perfiles de usuarios federados existentes. Este activador recibe los atributos sin procesar del proveedor de identidad como entrada y devuelve los atributos modificados que Amazon Cognito aplica al perfil de usuario.

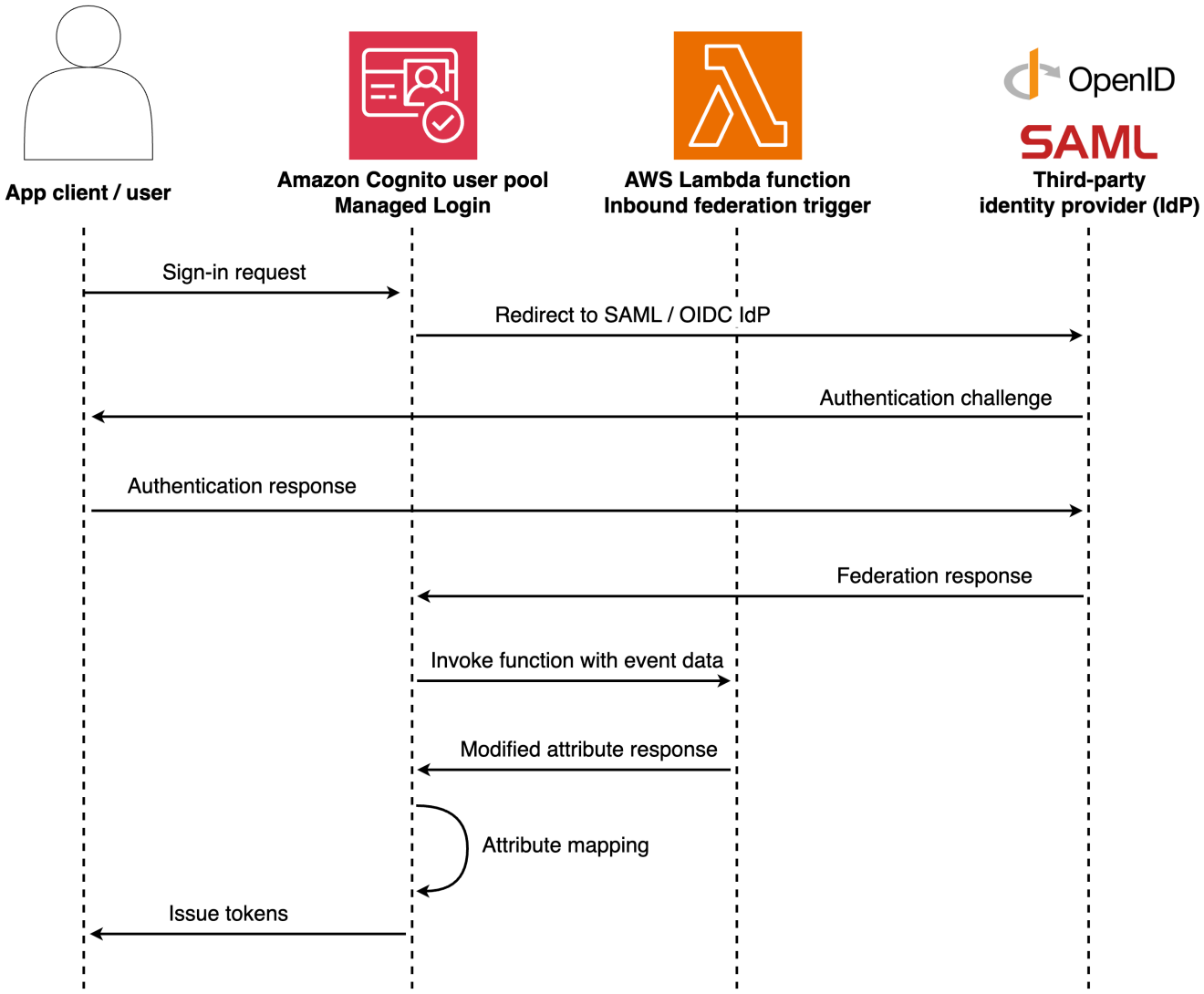
Temas

- Información general sobre flujos
- Parámetros de activación de Lambda de federación entrante
- Ejemplo de federación entrante: administración de miembros de grupos
- Ejemplo de federación entrante: truncar atributos grandes
- Ejemplo de federación entrante: registro de eventos de federación

Información general sobre flujos

Cuando un usuario se autentica con un proveedor de identidad externo, Amazon Cognito invoca el activador de federación entrante antes de crear o actualizar el perfil de usuario. El activador recibe

los atributos sin procesar del proveedor de identidad y puede transformarlos antes de que Amazon Cognito los almacene. Este flujo se produce tanto para los nuevos usuarios federados como para los usuarios existentes que vuelven a iniciar sesión mediante la federación.



Parámetros de activación de Lambda de federación entrante

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

```
{
```

```
    "version": "string",
    "triggerSource": "InboundFederation_ExternalProvider",
    "region": AWSRegion,
    "userPoolId": "string",
    "userName": "string",
    "callerContext": {
        "awsSdkVersion": "string",
        "clientId": "string"
    },
    "request": {
        "providerName": "string",
        "providerType": "string",
        "attributes": {
            "tokenResponse": {
                "access_token": "string",
                "token_type": "string",
                "expires_in": "string"
            },
            "idToken": {
                "sub": "string",
                "email": "string",
                "email_verified": "string"
            },
            "userInfo": {
                "email": "string",
                "given_name": "string",
                "family_name": "string"
            },
            "samlResponse": {
                "string": "string"
            }
        }
    },
    "response": {
        "userAttributesToMap": {
            "string": "string"
        }
    }
}
```

Parámetros de las solicitudes de federación entrantes

Nombre del proveedor

El nombre del proveedor de identidad externo.

Tipo de proveedor

El tipo de proveedor de identidad externo. Valores válidos: `OIDC`SAML,`Facebook`,`Google`,`SignInWithApple`,`LoginWithAmazon`.

attributes

Los atributos sin procesar recibidos del proveedor de identidad antes del procesamiento. La estructura varía según el tipo de proveedor.

Atributos. TokenResponse

Datos de respuesta del token OAuth desde el punto final. `/token` Disponible solo para proveedores sociales y de OIDC. Contiene`access_token`,`id_token`, `refresh_token` `token_type`e`expires_in`, y. `scope`

Attributes.idToken

JWT afirma que el token de identificación decodificado y validado. Disponible solo para el OIDC y los proveedores sociales. Contiene información de identidad de usuario verificada, que incluye `sub` (identificador de usuario único) `email`name,, `iss` (emisor), `aud` (audiencia), `exp` (caducidad) y `iat` (hora de emisión).

Attributes.UserInfo

Información de perfil de usuario ampliada desde el punto final. UserInfo Disponible solo para el OIDC y los proveedores sociales. Contiene atributos de perfil detallados`given_name`, como, `family_name` `picture`e`address`, y otros campos específicos del proveedor. Puede estar vacío si el IdP no es compatible con el UserInfo punto final o si la llamada al punto final falla.

Atributos. Respuesta SAML

Atributos de aserción de SAML. Disponible solo para proveedores de SAML. Contiene los atributos de la respuesta de SAML.

Parámetros de respuesta de la federación entrante

usuario AttributesToMap

Los atributos de usuario que se van a aplicar al perfil de usuario.

Important

Debe incluir TODOS los atributos de usuario que desee conservar en la respuesta, incluidos los atributos que no vaya a modificar. Los atributos no incluidos en la `userAttributesToMap` respuesta se eliminan y no se almacenan en el perfil de usuario. Esto se aplica tanto a los atributos modificados como a los no modificados.

Comportamiento de respuesta vacía

Si devuelves un objeto vacío `{}` para`userAttributesToMap`, todos los atributos originales del proveedor de identidad se conservan sin cambios. Esto actúa como una operación no operativa, como si la función Lambda nunca se hubiera ejecutado. Esto es diferente de omitir atributos, que los elimina.

Provider-specific atributos

La estructura de `request.attributes` varía en función de`providerType`. El OIDC y los proveedores sociales incluyen `tokenResponseidToken`, y `userInfo` objetos. Los proveedores de SAML incluyen solo el objeto. `samlResponse`

Ejemplo de federación entrante: administración de miembros de grupos

En este ejemplo, se muestra cómo asignar grupos de proveedores de identidad federados a grupos de grupos de usuarios de Amazon Cognito. Esta función extrae la pertenencia a un grupo de la respuesta federada y añade automáticamente los usuarios a los grupos de Amazon Cognito correspondientes, lo que elimina la necesidad de activar activadores posteriores a la autenticación.

Node.js

```
exports.handler = async (event) => {
    const { providerType, attributes } = event.request;

    // Extract user attributes based on provider type
    let userAttributesFromIdp = {};
    if (providerType === 'SAML') {
        userAttributesFromIdp = attributes.samlResponse || {};
    } else {
        // For OIDC and Social providers, merge userInfo and idToken
        userAttributesFromIdp = {
            ...(attributes.userInfo || {}),
            ...(attributes.idToken || {})
        };
    }

    // Extract groups from federated response
    const federatedGroups = userAttributesFromIdp.groups?.split(',') || [];

    // Map federated groups to Cognito groups
    const groupMapping = {
        'Domain Admins': 'Administrators',
        'Engineering': 'Developers',
        'Sales': 'SalesTeam'
    };

    // Filter to only in-scope groups
    const mappedGroups = federatedGroups
        .map(group => groupMapping[group.trim()])
        .filter(group => group); // Remove undefined values

    // Pass through attributes with mapped groups as custom attribute
    const attributesToMap = {
        ...userAttributesFromIdp,
        'custom:user_groups': mappedGroups.join(',')
    };

    // Remove original groups attribute
    delete attributesToMap.groups;

    event.response.userAttributesToMap = attributesToMap;
    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

```
{
    "userPoolId": "us-east-1_XXXXXXXXX",
    "request": {
        "providerName": "CorporateAD",
        "providerType": "SAML",
        "attributes": {
            "samlResponse": {
                "email": "jane.smith@company.com",
                "given_name": "Jane",
                "family_name": "Smith",
                "groups": "Engineering,Domain Admins",
                "department": "Engineering"
            }
        }
    },
    "response": {
        "userAttributesToMap": {}
    }
}
```

Ejemplo de federación entrante: truncar atributos grandes

En este ejemplo, se muestra cómo truncar los valores de los atributos que superan los límites de almacenamiento de Amazon Cognito. Esta función comprueba cada atributo del proveedor de identidad. Si el valor de un atributo supera los 2048 caracteres, trunca el valor y añade puntos suspensivos para indicar el truncamiento. Todos los demás atributos pasan sin cambios.

Node.js

```
exports.handler = async (event) => {
    const MAX_ATTRIBUTE_LENGTH = 2048;
```

```
    // Get the identity provider attributes based on provider type
    const { providerType, attributes } = event.request;
    let idpAttributes = {};

    if (providerType === 'SAML') {
        idpAttributes = attributes.samlResponse || {};
    } else {
        // For OIDC and Social providers, merge userInfo and idToken
        idpAttributes = {
            ...(attributes.userInfo || {}),
            ...(attributes.idToken || {})
        };
    }

    const userAttributes = {};

    // Process each attribute
    for (const [key, value] of Object.entries(idpAttributes)) {
        if (typeof value === 'string' && value.length > MAX_ATTRIBUTE_LENGTH) {
            // Truncate the value and add ellipsis
            userAttributes[key] = value.substring(0, MAX_ATTRIBUTE_LENGTH - 3) +
 '...';
            console.log(`Truncated attribute ${key} from ${value.length} to
 ${userAttributes[key].length} characters`);
        } else {
            // Keep the original value
            userAttributes[key] = value;
        }
    }

    // Return the modified attributes
    event.response.userAttributesToMap = userAttributes;
    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

```
{
    "version": "string",
    "triggerSource": "InboundFederation_ExternalProvider",
    "region": "us-east-1",
    "userPoolId": "us-east-1_XXXXXXXXX",
    "userName": "ExampleProvider_12345",
    "callerContext": {
        "awsSdkVersion": "string",
        "clientId": "string"
    },
    "request": {
        "providerName": "ExampleProvider",
        "providerType": "OIDC",
        "attributes": {
            "tokenResponse": {
                "access_token": "abcDE...",
                "token_type": "Bearer",
                "expires_in": "3600"
            },
            "idToken": {
                "sub": "12345",
                "email": "user@example.com"
            },
            "userInfo": {
                "email": "user@example.com",
                "given_name": "Example",
                "family_name": "User",
                "bio": "This is a very long biography that contains more than 2048
 characters..."
            }
        }
    },
    "response": {
        "userAttributesToMap": {}
    }
}
```

Ejemplo de federación entrante: registro de eventos de federación

En este ejemplo se muestra cómo registrar los eventos de autenticación federada para su supervisión y depuración. Esta función de ejemplo captura información detallada sobre los usuarios federados y sus atributos, lo que proporciona visibilidad del proceso de autenticación.

Node.js

```
exports.handler = async (event) => {
    const { providerName, providerType, attributes } = event.request;

    // Extract user attributes based on provider type
    let userAttributesFromIdp = {};
    if (providerType === 'SAML') {
        userAttributesFromIdp = attributes.samlResponse || {};
    } else {
        // For OIDC and Social providers, merge userInfo and idToken
        userAttributesFromIdp = {
            ...(attributes.userInfo || {}),
            ...(attributes.idToken || {})
        };
    }

    // Log federated authentication details
    console.log(JSON.stringify({
        timestamp: new Date().toISOString(),
        providerName,
        providerType,
        userEmail: userAttributesFromIdp.email,
        attributeCount: Object.keys(userAttributesFromIdp).length,
        attributes: userAttributesFromIdp
    }));

    // Pass through all attributes unchanged
    event.response.userAttributesToMap = userAttributesFromIdp;
    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

```
{
    "version": "string",
    "triggerSource": "InboundFederation_ExternalProvider",
    "region": "us-east-1",
    "userPoolId": "us-east-1_XXXXXXXXX",
    "userName": "CorporateAD_john.doe",
    "callerContext": {
        "awsSdkVersion": "string",
        "clientId": "string"
    },
    "request": {
        "providerName": "CorporateAD",
        "providerType": "SAML",
        "attributes": {
            "samlResponse": {
                "email": "john.doe@company.com",
                "given_name": "John",
                "family_name": "Doe",
                "department": "Engineering",
                "employee_id": "EMP12345"
            }
        }
    },
    "response": {
        "userAttributesToMap": {}
    }
}
```

Resultado esperado CloudWatch de los registros:

JSON

```
{
    "timestamp": "2025-01-14T21:17:40.153Z",
    "providerName": "CorporateAD",
    "providerType": "SAML",
    "userEmail": "john.doe@company.com",
```

```
    "attributeCount": 5,
    "attributes": {
        "email": "john.doe@company.com",
        "given_name": "John",
        "family_name": "Doe",
        "department": "Engineering",
        "employee_id": "EMP12345"
    }
}
```

Desencadenadores de Lambda de desafío de autenticación personalizado

Cuando cree flujos de autenticación para su grupo de usuarios de Amazon Cognito, puede querer que el modelo de autenticación no se limite a los flujos integrados. Normalmente, los desencadenadores de desafíos personalizados se suelen utilizar para implementar controles de seguridad adicionales, además del nombre de usuario, la contraseña y la autenticación multifactor (MFA). Un desafío personalizado es cualquier pregunta y respuesta que puedas generar en un lenguaje Lambda-supported de programación. Por ejemplo, puede que quiera solicitar a los usuarios que resuelvan un CAPTCHA o que respondan a una pregunta de seguridad antes de que se les permita autenticarse. También puede darse el caso de que necesite una integración con factores o dispositivos de autenticación especializados. O puede que ya haya desarrollado un software que autentique a los usuarios con una clave de seguridad de hardware o un dispositivo biométrico. La definición de autenticación correcta a un desafío personalizado es cualquier respuesta que su función de Lambda acepte como correcta; por ejemplo, una cadena fija o una respuesta satisfactoria de una API externa.

Puede iniciar la autenticación con el desafío personalizado y controlar el proceso de autenticación por completo, o puede realizar la autenticación con el nombre de usuario y la contraseña antes de que la aplicación reciba el desafío personalizado.

El desencadenador de Lambda de desafío de autenticación personalizado:

Define

Inicia una secuencia de desafío. Determina si desea iniciar un nuevo desafío, marcar la autenticación como completa o detener el intento de autenticación.

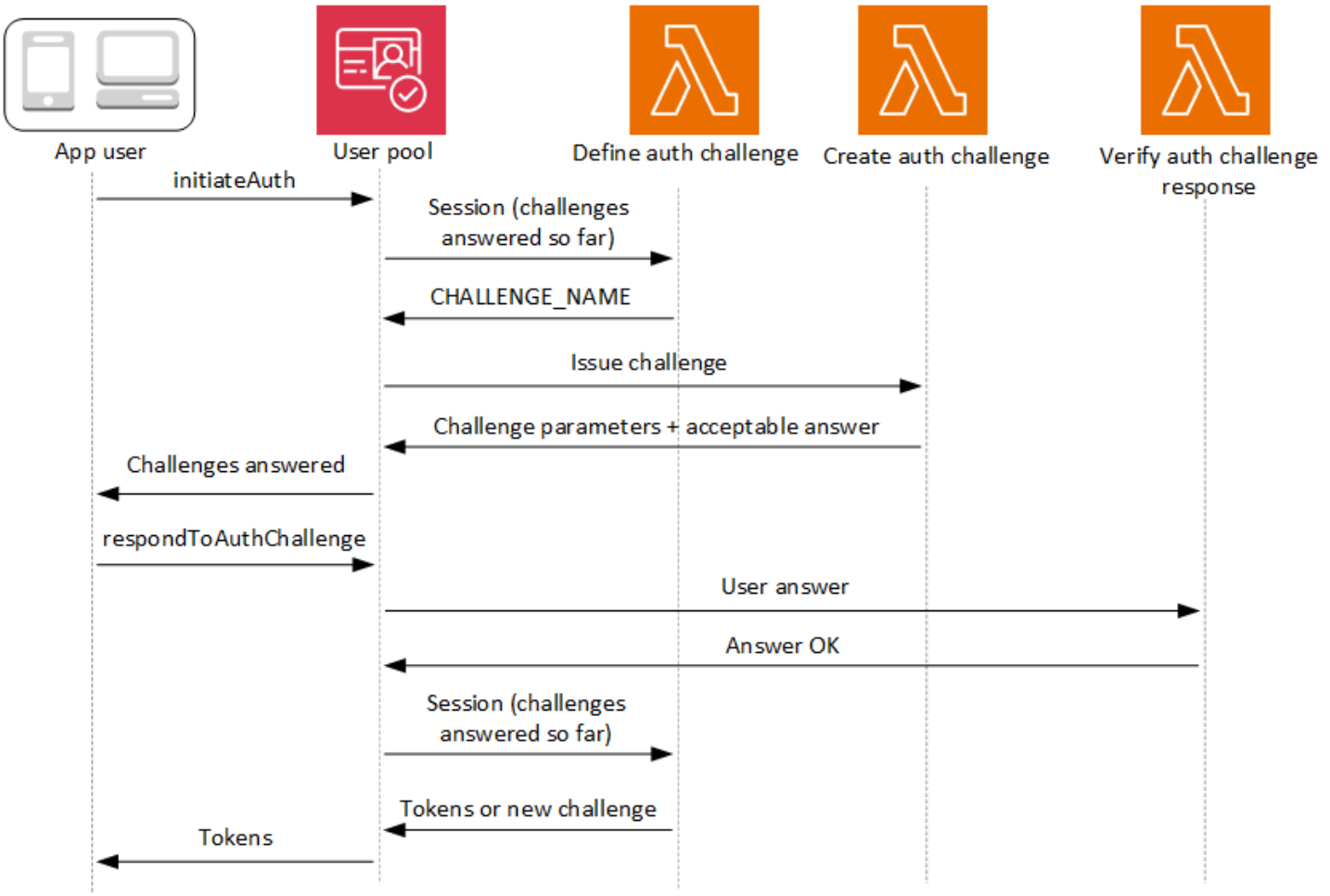
Crea

Remite a la aplicación la pregunta que el usuario debe responder. Esta función puede presentar una pregunta de seguridad o un enlace a un CAPTCHA que la aplicación debe mostrar al usuario.

Verifica

Conoce la respuesta esperada y la compara con la respuesta que proporciona la aplicación en la respuesta al desafío. La función puede llamar a la API de su servicio de CAPTCHA para recuperar el resultado esperado y compararla con la solución que el usuario propone.

Estas tres funciones de Lambda se encadenan para presentar un mecanismo de autenticación totalmente controlado y diseñado por usted. Como la autenticación personalizada requiere lógica de aplicación en el cliente y en las funciones de Lambda, no puede procesar la autenticación personalizada en el inicio de sesión administrado. Este sistema de autenticación requiere un esfuerzo adicional del desarrollador. La aplicación debe ejecutar el flujo de autenticación con la API de grupos de usuarios y gestionar el desafío resultante con una interfaz de inicio de sesión personalizada que sitúe la pregunta en el centro del desafío de autenticación personalizado.



Para obtener más información sobre cómo implementar una autenticación personalizada, consulte Flujo de autenticación personalizado y desafíos.

Autenticación entre las operaciones de la API InitiateAutho AdminInitiateAuth, y RespondToAuthChallengeo AdminRespondToAuthChallenge. 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. Una respuesta a un desafío puede ser un desafío nuevo. En dicho caso, la aplicación responde tantas veces como sea necesario a los nuevos desafíos. La autenticación se considerará correcta cuando la función de definición del desafío de autenticación analice los resultados obtenidos hasta el momento, determine que se han resuelto todos los problemas y devuelva `IssueTokens`.

> ¿Por qué no se activó tu activador de autenticación personalizado
>
> Si no se invoca la función de respuesta Crear desafío de autenticación o Verificar desafío de autenticación, la causa suele ser una de las siguientes:

- El cliente de la aplicación debe permitir un flujo de autenticación personalizado. Actívalo `ALLOW_CUSTOM_AUTH` (o, para un inicio de sesión basado en una elección`ALLOW_USER_AUTH`) en el cliente de la aplicación y comienza a iniciar sesión con el flujo correspondiente. Un grupo que solo permite la autenticación mediante SRP o contraseña nunca entra en la secuencia de desafíos personalizada.
- Amazon Cognito invoca las tres funciones en orden. Siempre llama primero a Definir desafío de autenticación y llama a Crear desafío de autenticación y Verify a desafío de autenticación solo cuando vuelve a aparecer la función Definir desafío de autenticación. `challengeName: CUSTOM_CHALLENGE` Si Create no se está ejecutando, confirma que Define retorna `challengeName: CUSTOM_CHALLENGE` con `issueTokens: false` y`failAuthentication: false`, en lugar de establecer o. `issueTokens: true` `failAuthentication: true`
- Los tres activadores deben estar asignados al grupo de usuarios. La falta de una tarea para alguno de ellos rompe la cadena.

Temas

- Autenticación SRP en flujos de desafíos personalizados
- 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

Autenticación SRP en flujos de desafíos personalizados

Puede hacer que Amazon Cognito verifique las contraseñas de los usuarios antes de que emita los desafíos personalizados. Todos los activadores Lambda asociados a la categoría de autenticación de las cuotas de tasa de solicitudes se ejecutan cuando se realiza 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.

A continuación se muestra un ejemplo de una solicitud de inicio `InitiateAuth` que precede a la autenticación personalizada con un flujo de SRP.

```
{
    "AuthFlow": "CUSTOM_AUTH",
    "ClientId": "1example23456789",
    "AuthParameters": {
        "CHALLENGE_NAME": "SRP_A",
        "USERNAME": "testuser",
        "SRP_A": "[SRP_A]",
        "SECRET_HASH": "[secret hash]"
    }
}
```

Restablecimiento de contraseñas en un flujo SRP de autenticación personalizada

Cuando los usuarios se encuentran en el estado `FORCE_CHANGE_PASSWORD`, el flujo de autenticación personalizado debe integrar el paso de cambio de contraseña y, al mismo tiempo, mantener la integridad de los desafíos de autenticación. Amazon Cognito invoca el desencadenador de Lambda Definir desafío de autenticación durante el desafío `NEW_PASSWORD_REQUIRED`. En este escenario, un usuario que inicie sesión con un flujo de desafío personalizado y una autenticación SRP puede establecer una nueva contraseña si se encuentra en un estado de restablecimiento de contraseña.

Cuando los usuarios se encuentran en el estado `RESET_REQUIRED` o `FORCE_CHANGE_PASSWORD` deben responder a un desafío `NEW_PASSWORD_REQUIRED` con un `NEW_PASSWORD`.

En la autenticación personalizada con SRP, Amazon Cognito devuelve un desafío `NEW_PASSWORD_REQUIRED` después de que los usuarios completen el desafío `PASSWORD_VERIFIER` SRP. El desencadenador Definir desafío de autenticación recibe los dos resultados del desafío en la matriz `session` y puede continuar con desafíos personalizados adicionales una vez que el usuario haya cambiado correctamente su contraseña.

El desencadenador Definir desafío de autenticación de Lambda debe gestionar la secuencia de desafíos mediante la autenticación SRP, el restablecimiento de la contraseña y los subsiguientes desafíos personalizados. El desencadenador recibe una serie de desafíos completados en el parámetro `session`, incluidos los resultados de `PASSWORD_VERIFIER` y `NEW_PASSWORD_REQUIRED`. Para ver un ejemplo de implementación, consulte Ejemplo de definición de desafíos de autenticación.

Pasos de un flujo de autenticación

Para los usuarios que necesitan verificar su contraseña antes de los desafíos personalizados, el proceso sigue estos pasos:

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 correctamente, ocurre una de estas dos cosas:

   Para los usuarios en estado normal:

   > Amazon Cognito llama a la función de Lambda con una nueva sesión que contiene `challengeName: PASSWORD_VERIFIER` y `challengeResult: true`.
   >
   > Para iniciar los desafíos personalizados, la función de Lambda responde con `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` y `failAuthentication: false`.

Para usuarios en el estado **`RESET_REQUIRED`** o **`FORCE_CHANGE_PASSWORD`**:

Amazon Cognito invoca a la función de Lambda con una sesión que contiene `challengeName: PASSWORD_VERIFIER` y `challengeResult: true`.

La función Lambda debería responder con `challengeName: NEW_PASSWORD_REQUIRED`, `issueTokens: false` y `failAuthentication: false`.

Tras cambiar correctamente la contraseña, Amazon Cognito invoca la función de Lambda con una sesión que contiene los resultados de `PASSWORD_VERIFIER` y `NEW_PASSWORD_REQUIRED`.

Para iniciar los desafíos personalizados, la función de Lambda responde con `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` y `failAuthentication: false`.

5. El bucle de desafíos se repite hasta que todos los desafíos tengan respuesta.

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`.

Administración de sesiones

El flujo de autenticación mantiene la continuidad de la sesión mediante una serie de identificadores de sesión y resultados de desafíos. Cada respuesta al desafío genera un nuevo identificador de sesión para evitar errores de reutilización de la sesión, lo cual es particularmente importante en los flujos de autenticación multifactor.

Los resultados del desafío se almacenan cronológicamente en la matriz de sesiones que reciben los desencadenadores Lambda. Para los usuarios en el estado `FORCE_CHANGE_PASSWORD`, la matriz de sesiones contiene lo siguiente:

1. `session[0]`: desafío `SRP_A` inicial
2. `session[1]`: desafío `PASSWORD_VERIFIER` inicial
3. `session[2]`: desafío `NEW_PASSWORD_REQUIRED` inicial
4. Elementos posteriores: resultados de desafíos personalizados adicionales

Ejemplo de flujo de autenticación

El siguiente ejemplo muestra un flujo de autenticación personalizado completo para un usuario en el estado `FORCE_CHANGE_PASSWORD` que debe completar tanto el cambio de contraseña como un desafío de CAPTCHA personalizado.

1. InitiateAuth request

```
{
    "AuthFlow": "CUSTOM_AUTH",
    "ClientId": "1example23456789",
    "AuthParameters": {
        "CHALLENGE_NAME": "SRP_A",
        "USERNAME": "testuser",
        "SRP_A": "[SRP_A]"
    }
}
```

2. InitiateAuth respuesta

```
{
    "ChallengeName": "PASSWORD_VERIFIER",
    "ChallengeParameters": {
        "USER_ID_FOR_SRP": "testuser"
    },
    "Session": "[session_id_1]"
}
```

3. RespondToAuthChallenge solicitud con **`PASSWORD_VERIFIER`**

```
{
    "ChallengeName": "PASSWORD_VERIFIER",
    "ClientId": "1example23456789",
    "ChallengeResponses": {
        "PASSWORD_CLAIM_SIGNATURE": "[claim_signature]",
        "PASSWORD_CLAIM_SECRET_BLOCK": "[secret_block]",
        "TIMESTAMP": "[timestamp]",
        "USERNAME": "testuser"
    },
    "Session": "[session_id_1]"
}
```

4. RespondToAuthChallenge respuesta con **`NEW_PASSWORD_REQUIRED`** desafío

```
{
    "ChallengeName": "NEW_PASSWORD_REQUIRED",
    "ChallengeParameters": {},
    "Session": "[session_id_2]"
}
```

5. RespondToAuthChallenge solicitud con **NEW_PASSWORD_REQUIRED**

```
{
    "ChallengeName": "NEW_PASSWORD_REQUIRED",
    "ClientId": "1example23456789",
    "ChallengeResponses": {
        "NEW_PASSWORD": "[password]",
        "USERNAME": "testuser"
    },
    "Session": "[session_id_2]"
}
```

6. RespondToAuthChallenge respuesta con un desafío personalizado de CAPTCHA

```
{
    "ChallengeName": "CUSTOM_CHALLENGE",
    "ChallengeParameters": {
        "captchaUrl": "url/123.jpg"
    },
    "Session": "[session_id_3]"
}
```

7. RespondToAuthChallenge solicitud con respuesta al desafío personalizado de CAPTCHA

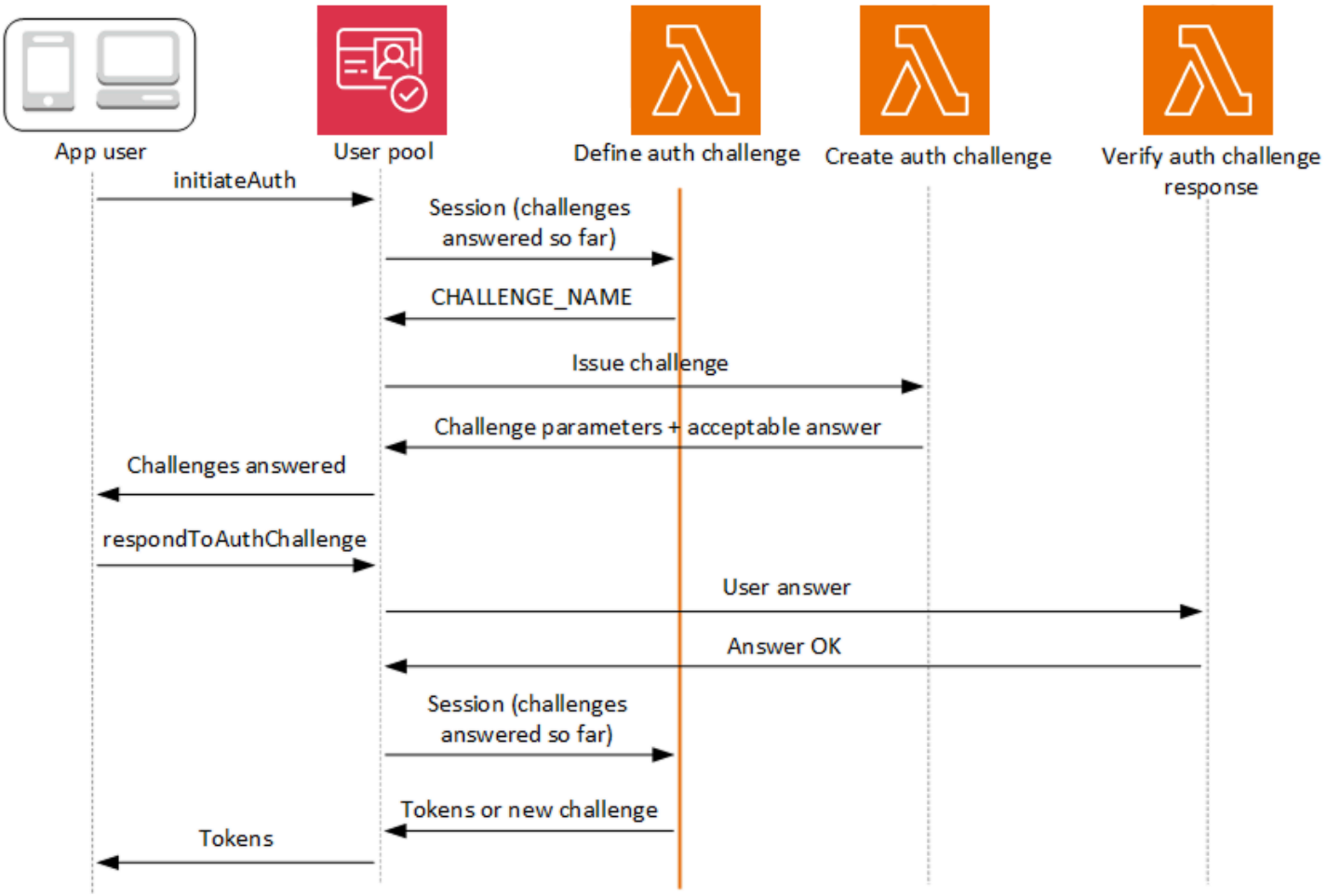
```
{
    "ChallengeName": "CUSTOM_CHALLENGE",
    "ClientId": "1example23456789",
    "ChallengeResponses": {
        "ANSWER": "123",
        "USERNAME": "testuser"
    },
    "Session": "[session_id_3]"
}
```

6. Respuesta final de éxito

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "IdToken": "eyJra789ghiEXAMPLE",
        "RefreshToken": "eyJjd123abcEXAMPLE",
        "TokenType": "Bearer"
    },
    "ChallengeParameters": {}
}
```

Desencadenador de Lambda para definir el desafío de autenticación

El desencadenador de definición de desafíos de autenticación es una función de Lambda que mantiene la secuencia de desafíos en un flujo de autenticación personalizado. Declara el éxito o el fracaso de la secuencia de desafíos y establece el siguiente desafío si la secuencia aún no se ha completado.



Definir desafío 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.

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.

usuario NotFound

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`, `EMAIL_OTP`, `SOFTWARE_TOKEN_MFA`, `DEVICE_SRP_AUTH`, `DEVICE_PASSWORD_VERIFIER` o `ADMIN_NO_SRP_AUTH`.

Cuando la función de definición de desafíos de autenticación emite un desafío `PASSWORD_VERIFIER` para un usuario que ha configurado la autenticación multifactor, Amazon Cognito lo continúa con un desafío `SMS_MFA`, `EMAIL_OTP` o `SOFTWARE_TOKEN_MFA`. Se trata de peticiones de código de autenticación multifactor. En su función, incluya la gestión de los eventos de entrada de los desafíos `SMS_MFA`, `EMAIL_OTP` y `SOFTWARE_TOKEN_MFA`. No necesita invocar ningún desafío de MFA desde la función de definición de desafíos de autenticación.

> **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. Para pasar estos datos a la función Lambda, puede usar el `ClientMetadata` parámetro en las operaciones AdminRespondToAuthChallengey RespondToAuthChallengeAPI. La solicitud que invoca la función de desafío de autenticación definida no incluye los datos transferidos en el ClientMetadata parámetro en AdminInitiateAuthlas operaciones de la API. 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. Cuando los usuarios completan la autenticación SRP con los desafíos `SRP_A` y `PASSWORD_VERIFIER`, esta función les pasa un `CUSTOM_CHALLENGE` que invoca al desencadenador Crear desafío de autenticación. Junto con nuestro ejemplo para Crear desafío de autenticación, esta secuencia ofrece un desafío de CAPTCHA para el desafío tres y una pregunta de seguridad para el desafío cuatro.

Una vez que el usuario resuelve el CAPTCHA y responde a la pregunta de seguridad, esta función confirma que su grupo de usuarios puede emitir tokens. La autenticación SRP no es necesaria; también puede configurar el CAPTCHA y la pregunta de seguridad como desafíos uno y dos. Si la función Definir desafío de autenticación no declara desafíos SRP, el éxito de los usuarios dependerá exclusivamente de sus respuestas a las solicitudes personalizadas.

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" &&
```

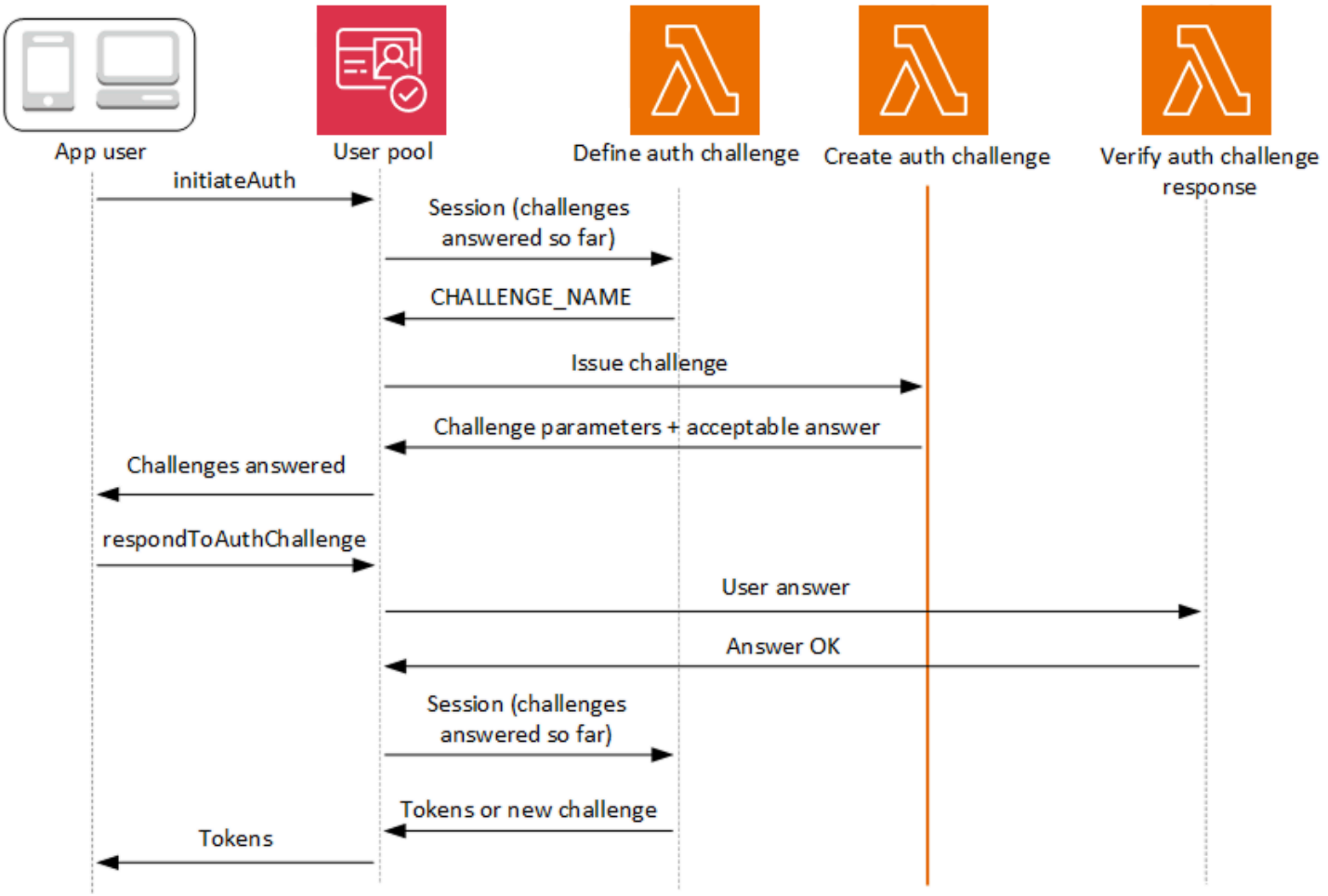
```
    event.request.session[2].challengeResult === true
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "CUSTOM_CHALLENGE";
  } else if (
    event.request.session.length === 4 &&
    event.request.session[3].challengeName === "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

El desencadenador de creación de desafíos de autenticación es una función de Lambda que contiene los detalles de todos los desafíos que el desencadenador de definición de desafíos de autenticación ha declarado. Procesa el nombre del desafío que el desencadenador de definición de desafíos de autenticación ha declarado y devuelve un valor `publicChallengeParameters` que la aplicación debe presentar al usuario. A continuación, esta función proporciona al grupo de usuarios la respuesta al desafío, `privateChallengeParameters`, que el grupo de usuarios pasa, a su vez, al desencadenador de verificación del desafío de autenticación. Mientras que el desencadenador de definición de desafíos de autenticación administra la secuencia del desafío, el desencadenador de creación de desafíos de autenticación administra el contenido del desafío.



Crear desafío 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 repite 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.

usuario NotFound

Este valor booleano se rellena cuando `PreventUserExistenceErrors` se establece como `ENABLED` en el cliente del grupo de usuarios.

challengeName

El nombre del nuevo desafío.

sesión

El elemento session consiste en una matriz de elementos de `ChallengeResult` que contienen, cada uno, los elementos siguientes:

challengeName

El tipo de desafío. Uno de los siguientes: `"CUSTOM_CHALLENGE"`, `"PASSWORD_VERIFIER"`, `"SMS_MFA"`, `"DEVICE_SRP_AUTH"`, `"DEVICE_PASSWORD_VERIFIER"`, `"NEW_PASSWORD_REQUIRED"` 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 usar el ClientMetadata parámetro en las acciones AdminRespondToAuthChallengey de la RespondToAuthChallengeAPI para pasar estos datos a la función Lambda. La solicitud que invoca la función de desafío de creación de autenticación no incluye los datos transferidos en el ClientMetadata parámetro en AdminInitiateAuthlas operaciones de la API. InitiateAuth

Parámetros de la respuesta para crear desafíos de autenticación

público ChallengeParameters

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.

privado ChallengeParameters

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

Esta función tiene dos desafíos personalizados que corresponden a la secuencia de desafíos de nuestro ejemplo de desafío con Definir desafío de autenticación. Los dos primeros desafíos son con autenticación SRP. Para el tercer desafío, esta función devuelve una URL de CAPTCHA a su aplicación en la respuesta al desafío. La aplicación renderiza el CAPTCHA en la URL indicada y devuelve los datos introducidos por 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.

Para el cuarto desafío, esta función devuelve una pregunta de seguridad. La aplicación renderiza la pregunta y solicita al usuario su respuesta. Cuando los usuarios resuelvan los dos desafíos personalizados, el desencadenador Definir desafío de autenticación confirma que su grupo de usuarios puede emitir tokens.

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";
  }

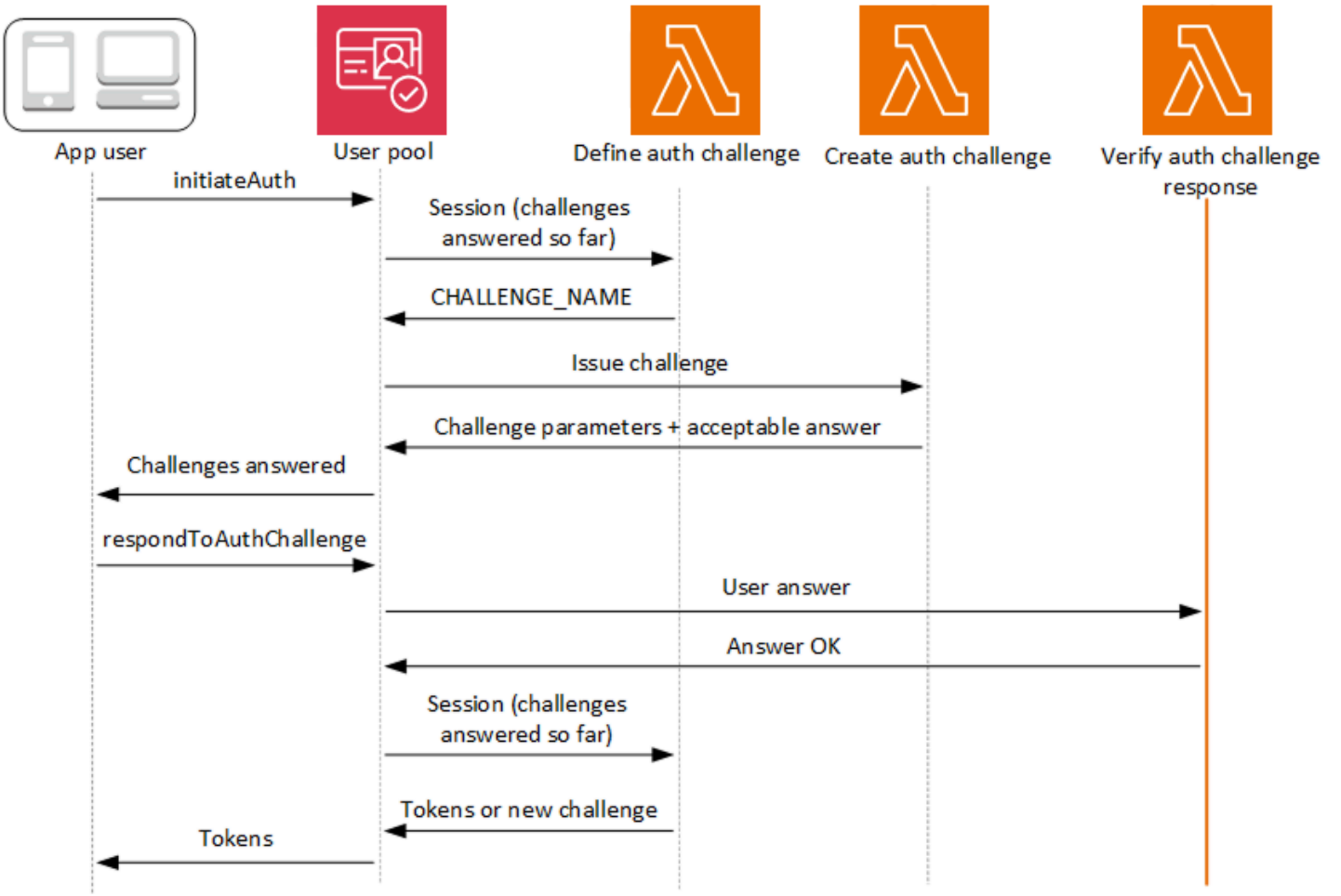
  if (event.request.session.length === 3) {
    event.response.publicChallengeParameters = {};
    event.response.privateChallengeParameters = {};
    event.response.publicChallengeParameters.securityQuestion =
      "Who is your favorite team mascot?";
    event.response.privateChallengeParameters.answer = "Peccy";
  }

  return event;
};

export { handler };
```

Desencadenador de Lambda para verificar la respuesta al desafío de autenticación

El desencadenador de verificación del desafío de autenticación es una función de Lambda que compara la respuesta proporcionada por el usuario con una respuesta conocida. Esta función indica al grupo de usuarios si el usuario ha respondido correctamente al desafío. Cuando el desencadenador de verificación del desafío de autenticación responde `true` a `answerCorrect`, la secuencia de autenticación puede continuar.



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.

usuario NotFound

Cuando Amazon Cognito establece `PreventUserExistenceErrors` en `ENABLED` para su cliente de grupo de usuarios, Amazon Cognito rellena este booleano.

privado ChallengeParameters

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. 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 pasa a la función de verificación de autenticación. AdminInitiateAuth

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 Verificar desafío de autenticación comprueba si la respuesta del usuario a un desafío coincide con la respuesta esperada. La respuesta del usuario se

define en función de las entradas de la aplicación y la respuesta preferida se define por `privateChallengeParameters.answer` en la respuesta del desencadenador Crear desafío de autenticación. Tanto la respuesta correcta como la respuesta dada forman parte del evento de entrada de esta función.

En este ejemplo, 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 el plan de funciones Essentials o Plus, puede generar la versión dos o el evento `V2_0` desencadenante con la personalización del token de acceso, y la versión tres o el evento `V3_0` desencadenante con la personalización del token de acceso para la autorización de máquina a máquina (M2M), incluida la concesión de credenciales de cliente y el funcionamiento de la API. `GetClientToken`

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` o `V3_0` es una solicitud única con los datos que Amazon Cognito escribiría en los tokens tanto de identidad y de acceso. Para personalizar ambos tokens, debe

actualizar la función para usar la versión dos o tres del desencadenador y enviar los datos de ambos tokens en la misma respuesta.

Amazon Cognito aplica las respuestas a eventos de la versión dos a los tokens de acceso de la autenticación de usuario, en la que un usuario humano ha presentado las credenciales a su grupo de usuarios. Las respuestas a los eventos de la versión tres se aplican a los tokens de acceso procedentes de la autenticación de usuario y de la autenticación automática, en la que los sistemas automatizados autorizan las solicitudes de token de acceso con los secretos del cliente de la aplicación. Aparte de las circunstancias de los tokens de acceso resultantes, los eventos de las versiones dos y tres son idénticos.

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 a su función de Lambda diferentes versiones de un evento desencadenante anterior a la generación del token. Un desencadenador `V1_0` proporciona los parámetros de modificación de los tokens de ID. Un desencadenador `V2_0` o `V3_0` proporciona parámetros para lo siguiente.

1. Las funciones de un desencadenador `V1_0`.
2. La posibilidad de personalizar los tokens de acceso.
3. La capacidad de transferir tipos de datos complejos y los valores de notificación de los tokens de ID y de acceso:
   - Cadena
   - Número
   - Booleano
   - Matriz de cadenas, números, valores booleanos o una combinación de cualquiera de ellos
   - JSON

> **Note**
>
> En el token de ID, puede rellenar con objetos complejos los valores de las notificaciones excepto `phone_number_verified`, `email_verified`, `updated_at` y `address`.

Los grupos de usuarios proporcionan de forma predeterminada eventos para `V1_0`. A fin de configurar el grupo de usuarios para enviar un evento de `V2_0`, elija la opción Versión del evento desencadenador en Características básicas + personalización del token de acceso para identidades de usuario al configurar el desencadenador en la consola de Amazon Cognito. Para producir eventos `V3_0`, elija Características básicas + personalización del token de acceso para las identidades de usuario y máquina. También puede establecer el valor de `LambdaVersion` en los LambdaConfigparámetros de una solicitud o de API. UpdateUserPool CreateUserPool Las versiones uno, dos y tres de los eventos están disponibles en los planes de características Essentials y Plus. Las operaciones M2M para los eventos de la tercera versión tienen una estructura de precios independiente de la fórmula de usuarios activos mensuales (MAU). Para obtener más información, consulte Precios de Amazon Cognito.

> **Note**
>
> Los grupos de usuarios que estaban operativos con la opción Características avanzadas de seguridad antes del 22 de noviembre de 2024 a las 18:00 GMT y que permanecen en el nivel de características Lite tienen acceso a las versiones uno y dos de los eventos del desencadenador Antes de la generación del token. Los grupos de usuarios de este nivel antiguo sin características de seguridad avanzadas tienen acceso a la primera versión del evento. La versión tres solo está disponible en Essentials y Plus.

Referencia sobre reclamaciones y ámbitos

Amazon Cognito limita las reclamaciones y los ámbitos que puede agregar, modificar o suprimir en los tokens de acceso e identidad. En la siguiente tabla, se describen las notificaciones que la función de Lambda puede y no puede modificar, así como los parámetros del evento desencadenador que afectan a la presencia o al valor de la notificación.

Reclamación	Tipo de token predeterminado	¿Puede añadir?	¿Puede modificar?	¿Puede suprimir	Parámetro de evento: añadir o modificar	Parámetro de evento: suprimir	Tipo de identidad	Versión del evento
Cualquier reclamación que no esté incluida en el esquema de token del grupo de usuarios	Ningun	Sí	Sí	N/A	`claimsToAddOrOverride`	`claimsToSuppress`	Usuario, máquina [1]	Todos [2]
`scope`	Accesc	Sí	Sí	Sí	`scopesToAdd`	`scopesToSuppress`	Usuario, máquina [1]	`v2_0`, `v3_0`
`cognito:groups`	ID, acceso	Sí	Sí	Sí	`groupsToOverride`	`claimsToSuppress`	Usuario	Todos [2]
`cognito:preferred_role`	ID	Sí	Sí	Sí	`preferredRole`	`claimsToSuppress` [3]	Usuario	Todos
`cognito:roles`	ID	Sí	Sí	Sí	`iamRolesToOverride`	`claimsToSuppress` [3]	Usuario	Todos
`cognito:username`	ID	No	No	No	N/A	N/A	Usuario	N/A
Cualquier otra	Ningun	No	No	No	N/A	N/A	N/A	N/A

Reclamación	Tipo de token predeterminado	¿Puede añadir?	¿Puede modificar?	¿Puede suprimir?	Parámetro de evento: añadir o modificar	Parámetro de evento: suprimir	Tipo de identidad	Versión del evento
reclamación con prefijo `cognito:`								
`username`	Acceso	No	No	No	N/A	N/A	Usuario	`v2_0`, `v3_0`
`sub`	ID, acceso	No	No	No	N/A	N/A	Usuario	N/A
atributo OIDC estándar	ID	Sí	Sí	Sí	`claimsToAddOrOverride`	`claimsToSuppress`	Usuario	Todos
Atributo `custom:`	ID	Sí	Sí	Sí	`claimsToAddOrOverride`	`claimsToSuppress`	Usuario	Todos
Atributo `dev:`	ID	No	No	Sí	N/A	`claimsToSuppress`	Usuario	Todos
`identities`	ID	No	No	No	N/A	N/A	Usuario	N/A
`aud`[4]	ID	No	No	No	N/A	N/A	Usuario, máquina	N/A
`client_id`	Acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`event_id`	Acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A

Reclamación	Tipo de token predeterminado	¿Puede añadir?	¿Puede modificar?	¿Puede suprimir?	Parámetro de evento: añadir o modificar	Parámetro de evento: suprimir	Tipo de identidad	Versión del evento
`device_key`	Acceso	No	No	No	N/A	N/A	Usuario	N/A
`version`	Acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`acr`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`amr`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`at_hash`	ID	No	No	No	N/A	N/A	Usuario, máquina	N/A
`auth_time`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`azp`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`exp`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`iat`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`iss`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`jti`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A

Reclamación	Tipo de token predeterminado	¿Puede añadir?	¿Puede modificar?	¿Puede suprimir	Parámetro de evento: añadir o modificar	Parámetro de evento: suprimir	Tipo de identidad	Versión del evento
`nbf`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`nonce`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`origin_jti`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A
`token_use`	ID, acceso	No	No	No	N/A	N/A	Usuario, máquina	N/A

[1] Los tokens de acceso para las identidades de las máquinas solo están disponibles con la versión `v3_0` del evento de entrada del desencadenador. La versión tres solo está disponible en los niveles de características Essentials y Plus. Los grupos de usuarios del nivel Lite pueden recibir eventos `v1_0`. Los grupos de usuarios del nivel Lite con características de seguridad avanzadas pueden recibir eventos `v1_0` y `v2_0`.

[2] Configure su desencadenador Antes de la generación del token en la versión de eventos `v1_0` solo para el token de ID, en la versión `v2_0` para el token de acceso e ID y en la versión `v3_0` para el token de acceso e ID con capacidades para las identidades de máquinas.

[3] Para suprimir las reclamaciones `cognito:preferred_role` y `cognito:roles`, añada `cognito:groups` a `claimsToSuppress`.

[4] Puede agregar una reclamación de `aud` para acceder a los tokens, pero el valor debe coincidir con el ID de cliente de 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

En todas las versiones de eventos del desencadenador de Lambda Antes de la generación del token, puede personalizar el contenido de un token de identidad (ID) desde su 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 Descripción del token de identidad (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

En las versiones de eventos dos y tres del desencadenador de Lambda Antes de la generación del token, puede personalizar el contenido de un token de acceso desde su 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. Para la autorización de máquina a máquina (M2M), ya sea que el token de acceso provenga de una concesión de credenciales de cliente o de la operación de la `GetClientToken` API, Amazon Cognito solo invoca el activador previo a la generación del token cuando el grupo de usuarios está configurado para un evento de la versión tres (). `V3_0` Para obtener más información acerca de los tokens de acceso, consulte Descripción del token de acceso.

Los usos del desencadenador de Lambda previo a la generación de tokens con un token de acceso incluyen los siguientes.

- Añada o suprima los ámbitos 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.

Consola de administración de AWS

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. Seleccione el menú Extensiones y localice los desencadenadores de Lambda.
4. Agregue o edite un desencadenador previo a la generación de tokens.
5. Elija una función de Lambda en Asignar función de Lambda.
6. Elija una versión de evento desencadenador Características básicas + personalización de token de acceso para identidades de usuario o Características básicas + personalización de token de acceso para identidades de usuario y máquina. 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 solicitud de API o una solicitud de API. CreateUserPoolUpdateUserPool
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 y del cliente de aplicación.

Incluya el siguiente contenido en el parámetro `LambdaVersion` de la solicitud. El valor `V2_0` en `LambdaVersion` hace que el grupo de usuarios agregue parámetros para los tokens de acceso y que aplique los cambios en estos. El valor `V3_0` en `LambdaVersion` produce el mismo evento que `V2_0`, pero hace que su grupo de usuarios también aplique cambios a los tokens de acceso M2M. 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": "V3_0"
},
```

Metadatos del cliente para credenciales de cliente máquina a máquina (M2M)

Puede pasar los metadatos del cliente en las solicitudes M2M. Los metadatos del cliente son información adicional de un entorno de usuario o aplicación que puede contribuir a los resultados de una Desencadenador de Lambda anterior a la generación del token. En las operaciones de autenticación con un usuario principal, puedes pasar los metadatos del cliente al activador previo a la generación del token en el cuerpo de AdminRespondToAuthChallengelas solicitudes a la RespondToAuthChallengeAPI. Dado que las aplicaciones dirigen el flujo de generación de tokens de acceso para M2M con solicitudes directas al Punto de conexión de token, tienen un modelo diferente. En el cuerpo POST de las solicitudes de token de credenciales de cliente, pasa un `aws_client_metadata` parámetro con el objeto de metadatos del cliente URL-encoded (`x-www-form-urlencoded`) a la cadena. Para obtener una solicitud de ejemplo, consulte Credenciales de cliente con autorización básica. A continuación, puede ver un ejemplo de parámetro que transfiere los pares clave-valor `{"environment": "dev", "language": "en-US"}`.

```
aws_client_metadata=%7B%22environment%22%3A%20%22dev%22,%20%22language%22%3A%20%22en-US%22%7D
```

Más recursos

- How to customize access tokens in Amazon Cognito user pools

Temas

- Fuentes del desencadenador de Lambda de pregeneración de tokens
- Parámetros del desencadenador de Lambda de pregeneración de tokens
- Ejemplo de la segunda versión de un evento desencadenante previo al token: añadir y suprimir notificaciones, ámbitos y grupos
- Ejemplo de la versión dos de evento de generación anterior al token: añadir notificaciones con objetos complejos
- Ejemplo uno de versión de evento de generación anterior al token: Agregar una notificación nueva y suprimir otra existente

- Ejemplo uno de versión de evento de generación anterior al token: Modificar la pertenencia de un usuario a un grupo

Fuentes del desencadenador de Lambda de pregeneración de tokens

Valor de triggerSource	Event
`TokenGeneration_HostedAuth`	Se llama durante la autenticación desde la página de inicio de sesión con el inicio de sesión administrado 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_ClientCredentials`	Se llama después de la concesión de credenciales de un cliente M2M. Su grupo de usuarios solo envía este evento cuando su versión de evento es `V3_0`.
`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 one

El token de la versión uno puede establecer la pertenencia a grupos, los roles de IAM y las nuevas notificaciones en los tokens de ID. La anulación de la pertenencia a un grupo también se aplica a la reclamación `cognito:groups` de los tokens de acceso.

```
{
    "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"
            }
```

```
        }
    }
}
```

Versions two and three

Las versiones dos y tres solicitan eventos que añaden campos que personalizan el token de acceso. Los grupos de usuarios aplican los cambios de los eventos de la tercera versión a los tokens de acceso para las identidades de las máquinas. Estas versiones también añaden compatibilidad con tipos de datos `claimsToOverride` complejos en el objeto de respuesta. La función de 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 ellos
- 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

Name	Description (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
gruposToOverride	Los grupos del grupo de usuarios de los que es miembro su usuario.	1
iamRolesToOverride	Puede asociar un grupo de grupos de usuarios a un rol AWS Identity and Access Management (IAM). Este elemento es una lista de todos los roles de IAM de los grupos a los que pertenece su usuario.	1

Name	Description (Descripción)	Versión mínima del evento del desencadenador
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 InitiateAuthAPI en AdminInitiateAuthla solicitud que transfiere a la función de generación previa del token.	1
alcances	Acceda a ámbitos de tokens. 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

Name	Description (Descripción)	Versión mínima del evento del desencadenador
reclamaciones OverrideDetails	Un contenedor para todos los elementos de un evento desencadenante `V1_0`.	1

Name	Description (Descripción)	Versión mínima del evento del desencadenador
reclamaciones AndScopeO verrideDetails	Un contenedor para todos los elementos de un evento desencadenador `V2_0` o `V3_0`.	2
id TokenGene ration	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 o superior, pero los elementos secundarios aparecen en los eventos de la versión 1.	2
acceso TokenGeneration	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 y superior.	2
reclamaciones ToAddOrOv erride	Un mapa de una o más reclamaciones y los valores que desee agregar o modificar. Para las reclamaci ones relacionadas con el grupo, utilice `groupOver rideDetails` en su lugar. En los eventos de la versión 2 y superior, este elemento aparece en `accessTokenGeneration` y `idTokenGe neration` .	1 * -
reclamaciones ToSuppress	Una lista de reclamaciones que quiere que Amazon Cognito suprima. Si tu función suprime y reemplaza un valor de notificación, Amazon Cognito suprime la notificac ión. En los eventos de la versión 2 y superior, este elemento aparece en `accessTokenGeneration` y `idTokenGe neration` .	1

Name	Description (Descripción)	Versión mínima del evento del desencadenador
grupo OverrideDetails	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. Las anulaciones de grupo son los únicos cambios en el token de acceso que pueden realizar los eventos de la versión 1.	1
alcances ToAdd	Una lista de ámbitos 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
osciloscopios ToSuppress	Una lista de ámbitos 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 uno pueden devolver cadenas. Los objetos de respuesta a los eventos de la versión dos y tres 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 versión dos de evento de generación anterior al token: añadir notificaciones con objetos complejos

En este ejemplo, se realizan las siguientes modificaciones a los tokens de un usuario.

1. Agrega notificaciones de tipo numérico, cadena, booleano y JSON al token de ID. Este es el único cambio que los eventos desencadenantes de la versión dos ponen a disposición del token de ID.
2. Agrega notificaciones de tipo numérico, cadena, booleano y JSON al token de acceso.
3. Añade tres ámbitos al token de acceso.
4. Suprime la reclamación `email` en los tokens de acceso e ID.
5. Suprime el ámbito `aws.cognito.signin.user.admin` en el 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"]
            },
            "accessTokenGeneration": {
                "claimsToAddOrOverride": claims,
                "claimsToSuppress": ["email"],
                "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 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
- Parámetros del desencadenador de Lambda para migrar usuarios
- Ejemplo: Migrar un usuario con una contraseña existente
- Solución de problemas del desencadenante de la migración del usuario

Fuentes del desencadenador de Lambda para migrar usuarios

Valor de triggerSource	Event
`UserMigration_Authentication` [1]	Migración de usuarios al iniciar sesión.
`UserMigration_ForgotPassword`	Migración de usuarios durante el flujo de recuperación de contraseñas olvidadas.

[1] Amazon Cognito no invoca este desencadenador cuando los usuarios se autentican con un inicio de sesión sin contraseña.

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.

contraseña

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. Para pasar estos datos a la función Lambda, puede usar el ClientMetadata parámetro en las acciones InitiateAuthy de la AdminInitiateAuthAPI.

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. Para pasar estos

datos a la función Lambda, puede usar el ClientMetadata parámetro en las acciones AdminRespondToAuthChallengey de la ForgotPasswordAPI.

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 su grupo de usuarios con atributos de alias además del nombre de usuario para iniciar sesión y el usuario ha introducido un valor de alias válido como nombre de usuario. Este valor del alias puede ser una dirección de correo electrónico, un nombre de usuario preferido o un número de teléfono.

Atributos	Requisito
	Si la solicitud y el grupo de usuarios cumplen los requisitos de alias, la respuesta de su función debe asignar el `username` parámetro que recibió a un atributo de alias. Además, la respuesta debe asignar su propio valor al `username` atributo. Si el grupo 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.

final UserStatus

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. Luego, si la contraseña no cumple con la política, UserStatus defina final en`RESET_REQUIRED`.

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.

deseado DeliveryMediums

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.

fuerza AliasCreation

Si estableces este parámetro en, `TRUE` y el número de teléfono o la dirección de correo electrónico del UserAttributes parámetro ya existen como alias con un usuario diferente, la llamada a la API migra 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

```
exports.handler = (event, context, callback) => {
  var user;
```

```
  if (event.triggerSource == "UserMigration_Authentication") {
    // authenticate the user with your existing user directory service
    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";
      context.succeed(event);
    } else {
      // Return error to Amazon Cognito
      callback("Bad password");
    }
  } else if (event.triggerSource == "UserMigration_ForgotPassword") {
    // Lookup the user in your existing user directory service
    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";
      context.succeed(event);
    } else {
      // Return error to Amazon Cognito
      callback("Bad password");
    }
  } else {
    // Return error to Amazon Cognito
    callback("Bad triggerSource " + event.triggerSource);
  }
};
```

Solución de problemas del desencadenante de la migración del usuario

Si la migración no funciona según lo esperado aunque la función devuelva la respuesta documentada, compruebe las siguientes causas comunes.

No se invoca el disparador

Amazon Cognito invoca el activador de migración de usuarios solo cuando el usuario no existe aún en el grupo de usuarios. Solo se aplica a los flujos de inicio de sesión `USER_PASSWORD_AUTH`, `ADMIN_USER_PASSWORD_AUTH`, y `USER_AUTH` (con contraseña) y al flujo de contraseña olvidada. No se invoca para iniciar sesión con SRP (`USER_SRP_AUTH`), porque la contraseña del usuario nunca llega a tu función en ese flujo, ni para iniciar sesión sin contraseña. Si el activador no se activa, confirma que el cliente de la aplicación permite uno de estos flujos y que el usuario aún no está presente en el grupo.

Amazon Cognito no puede invocar la función

Si configura el activador con una política basada en recursos en la función y no a través de la consola de Amazon Cognito, la política debe permitir que el director `cognito-idp.amazonaws.com` del servicio `lambda:InvokeFunction` llame, según el ARN de su grupo de usuarios. La consola le añade este permiso; una función conectada con la infraestructura como código o la API de Lambda necesita que se añada de forma explícita. Sin él, la migración falla porque Amazon Cognito no puede invocar el desencadenador.

El usuario inicia sesión pero no se crea ningún perfil

Amazon Cognito crea el usuario migrado solo cuando la función devuelve la respuesta con la forma exacta que esperaba: el `userAttributes` campo debe estar presente e incluir todos los atributos marcados como obligatorios en el grupo de usuarios. Una función que autentica al usuario pero devuelve un `response` objeto vacío o con un formato incorrecto, u omite un atributo obligatorio, completa la llamada de inicio de sesión sin conservar un perfil. Vuelva `userAttributes` con todos los atributos necesarios y establezca y según sea necesario. `finalUserStatus` `messageAction`

Desencadenador de Lambda para mensajes personalizados

Si tiene un estándar externo para los mensajes de correo electrónico y SMS que desea enviar a los usuarios, o si desea aplicar su propia lógica en tiempo de ejecución al formato de los mensajes de los usuarios, añada un desencadenador de mensajes personalizado al grupo de usuarios. El Lambda de mensaje personalizado recibe el contenido de todos los mensajes de correo electrónico y SMS antes de que el grupo de usuarios los envíe. En ese momento, la función de Lambda tiene la oportunidad de modificar el contenido y el asunto del mensaje.

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.

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**
>
> El evento de entrada para una función de Lambda de mensaje personalizado con el origen del desencadenador `CustomMessage_AdminCreateUser` incluye 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 tu función debe incluir variables de marcador de posición para el nombre de usuario y el código. Los marcadores de posición del mensaje son los valores de `request.usernameParameter` y `request.codeParameter`. Estos valores suelen ser `{username}` y `{####}`; como práctica recomendada, haga referencia a los valores de entrada en lugar de codificar de forma rígida los nombres de las variables.

> **Note**
>
> Los mensajes que configure en el menú de plantillas de mensajes de la consola de Amazon Cognito pueden anular el cuerpo que defina la función de mensajes personalizada. Si el mensaje de su función no aparece, compruebe si la plantilla de mensajes de la consola está configurada para el mismo evento. Para obtener más información, consulte Configuración de los mensajes de verificación, invitación, autenticación y MFA.

Temas

- Fuentes de desencadenadores de Lambda para mensajes personalizados
- Parámetros de desencadenadores de Lambda para mensajes personalizados
- Ejemplo de mensaje personalizado para registrarse
- Ejemplo de mensaje personalizado para la creación de usuarios por parte del administrador

Fuentes de desencadenadores de Lambda para mensajes personalizados

Valor de triggerSource	Event
`CustomMessage_SignUp`	Mensaje personalizado para enviar el código de confirmación posterior a la inscripción.
`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 mensajes personalizada no incluye los

datos transferidos en el ClientMetadata parámetro en AdminInitiateAuthlas operaciones de la 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 UTF-8 caracteres. Esta longitud incluye el código de verificación. Puede utilizar etiquetas HTML en estos mensajes de correo electrónico.

La longitud máxima de los mensajes SMS es de 140 UTF-8 caracteres. Esta longitud incluye el código de verificación.

> **Note**
>
> Si la función devuelve un mensaje de correo electrónico de más de 20 000 UTF-8 caracteres, Amazon Cognito no devuelve ningún error. En su lugar, descarta silenciosamente el mensaje personalizado y envía el mensaje predeterminado para ese evento. Si tu mensaje personalizado no llega a los usuarios, comprueba primero su longitud teniendo en cuenta este límite.

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",
 "region": "us-west-2",
 "userPoolId": "us-west-2_EXAMPLE",
 "userName": "test-user",
 "callerContext": {
  "awsSdkVersion": "aws-sdk-unknown-unknown",
  "clientId": "1example23456789"
 },
 "triggerSource": "CustomMessage_SignUp",
 "request": {
  "userAttributes": {
   "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
   "cognito:user_status": "CONFIRMED",
   "email_verified": "true",
   "phone_number_verified": "true",
   "phone_number": "+12065551212",
   "email": "test-user@example.com"
  },
  "codeParameter": "{####}",
```

```
  "linkParameter": "{##Click Here##}",
  "usernameParameter": "None"
 },
 "response": {
  "smsMessage": "None",
  "emailMessage": "None",
  "emailSubject": "None"
 }
}
```

Ejemplo de mensaje personalizado para la creación de usuarios por parte del administrador

La solicitud que Amazon Cognito ha enviado a este ejemplo de función de Lambda de mensajes personalizados tiene un valor `triggerSource` de `CustomMessage_AdminCreateUser`, y un nombre de usuario y una contraseña temporal. La función rellena `${event.request.codeParameter}` con la contraseña temporal de la solicitud y `${event.request.usernameParameter}` con el nombre de usuario de la solicitud.

Los mensajes personalizados deben insertar los valores `codeParameter` y `usernameParameter` en `smsMessage` y `emailMessage`, en el objeto 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 UTF-8 caracteres. Esta longitud incluye el código de verificación. Puede usar etiquetas HTML en estos correos electrónicos. La longitud máxima de los mensajes SMS es de 140 UTF-8 caracteres. Esta longitud incluye el código de verificación.

> **Note**
>
> Si la función devuelve un mensaje de correo electrónico de más de 20 000 UTF-8 caracteres, Amazon Cognito no devuelve ningún error. En su lugar, descarta silenciosamente el mensaje personalizado y envía el mensaje predeterminado para ese evento. Si tu mensaje personalizado no llega a los usuarios, comprueba primero su longitud teniendo en cuenta este límite.

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 desencadenadores de Lambda `CustomEmailSender` y `CustomSMSSender` admiten notificaciones por correo electrónico y SMS de terceros en grupos de usuarios. 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 envía a los usuarios invitaciones, códigos de MFA, códigos de confirmación, códigos de verificación y 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 gestionada por el AWS KMS cliente y el. AWS Encryption SDK AWS Encryption SDK Se trata de una biblioteca de cifrado del lado del cliente que le ayuda a cifrar y descifrar datos genéricos.

CustomEmailSender

Amazon Cognito invoca este desencadenador para enviar notificaciones por correo electrónico a los usuarios.

PersonalizadaSMSSender

Amazon Cognito invoca este desencadenador para enviar notificaciones por SMS a los usuarios.

Conceptos sobre el cifrado

Amazon Cognito no envía los códigos de los usuarios en texto sin formato en los eventos que envía a desencadenadores de remitentes personalizados. Las funciones de Lambda deben descifrar códigos en los eventos. Los siguientes conceptos forman la arquitectura de cifrado que la función debe utilizar para obtener los códigos que puede entregar a los usuarios.

AWS KMS

AWS KMS es un servicio gestionado para crear y controlar claves. AWS KMS Estas claves cifran los datos. Para obtener más información, consulte ¿Qué es AWS Key Management Service?.

Clave de 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 usar una clave KMS simétrica, debe llamar AWS KMS. Amazon Cognito utiliza claves simétricas. La misma clave cifra y descifra. Para obtener más información, consulte Claves KMS simétricas.

Conceptos importantes sobre los desencadenadores de Lambda Remitente personalizado

- 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.

  La operación `UpdateUserPool` establece la configuración de Lambda. Las solicitudes para esta operación requieren todos los parámetros de su grupo de usuarios y los parámetros que desea cambiar. Si no proporciona todos los parámetros relevantes, Amazon Cognito establece los valores de los parámetros que faltan en sus valores predeterminados. Como se muestra en el siguiente ejemplo de AWS CLI, incluya entradas para todas las funciones de Lambda que desee añadir o conservar en su grupo de usuarios. Para obtener más información, consulte Actualización de la configuración del grupo de usuarios y del cliente de aplicación.

```
   #Send this parameter in an 'aws cognito-idp update-user-pool' CLI command,
including any existing
   #user pool configurations. This snippet also includes a pre sign-up trigger for
syntax reference. The pre sign-up trigger
   #doesn't have a role in custom sender triggers.
```

```
      --lambda-config "PreSignUp=lambda-arn, \
                       CustomSMSSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
                       CustomEmailSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
                       KMSKeyID=key-id"
```

Para las solicitudes que utilizan el cuerpo JSON de `UpdateUserPool`, el siguiente fragmento de `LambdaConfig` asigna funciones personalizadas de envío por SMS y correo electrónico.

```
"LambdaConfig": {
   "KMSKeyID": "arn:aws:kms:us-
east-1:111122223333:key/a6c4f8e2-0c45-47db-925f-87854bc9e357",
   "CustomEmailSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:111122223333:function:MyFunction",
      "LambdaVersion": "V1_0"
   },
   "CustomSMSSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:111122223333:function:MyFunction",
      "LambdaVersion": "V1_0"
   }
```

- Para eliminar un activador Lambda de remitente personalizado con un `update-user-pool` AWS CLI comando, 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.
- 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.

Activación de desencadenadores de Lambda Remitente personalizado

A fin de utilizar lógica personalizada para enviar mensajes SMS o correos electrónicos a su grupo de usuarios, configure los desencadenadores Remitente personalizado. 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 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 del comportamiento predeterminado, que envía un mensaje por correo o SMS.

1. Cree una clave de cifrado simétrica en AWS Key Management Service ().AWS KMS Amazon Cognito genera secretos (contraseñas temporales, códigos de verificación, contraseñas de autenticación de un solo uso y códigos de confirmación) y, a continuación, utiliza esta clave KMS para cifrar los secretos con AWS Encryption SDK. A continuación, puede utilizar AWS Encryption SDK la función Lambda para descifrar los secretos y enviarlos al usuario en texto plano.
2. La entidad principal de IAM que crea o actualiza el grupo de usuarios crea una concesión única con la clave de KMS que Amazon Cognito utiliza para cifrar el código. Otorgue estos permisos `CreateGrant` de entidad principal a su clave de KMS. Para que este ejemplo de política de claves KMS sea efectiva, el administrador que actualice el grupo de usuarios debe iniciar sesión con una sesión de rol asumido para el rol de IAM `arn:aws:iam::111222333444:role/my-example-administrator-role`.

   Aplique la siguiente política basada en recursos, modificada para su entorno, a su clave KMS.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/my-example-administrator-role"
        },
        "Action": "kms:CreateGrant",
        "Resource": "arn:aws:kms:us-west-2:111122223333:key/1example-2222-3333-4444-999example",
        "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:userpool-id": "us-west-2_EXAMPLE"
            }
```

```
        }
    },
    {
        "Sid": "Allow Lambda to decrypt",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/my-lambda-function-role"
        },
        "Action": "kms:Decrypt",
        "Resource": "*"
    }]
}
```

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.

    a. Asigne un rol de ejecución de Lambda que tenga, como mínimo, permisos `kms:Decrypt` para su clave KMS.

    b. Elabore el código de su función de Lambda para enviar sus mensajes. El evento de entrada de su función contiene un secreto. En su función, descifre el secreto con el AWS Encryption SDK y procese los metadatos relevantes. A continuación, envíe el código, su propio mensaje personalizado y el número de teléfono de destino a la API personalizada que entrega el mensaje.

    c. 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.

        i. Exporte su función de Lambda como un archivo .zip en la Consola de administración de AWS.

        ii. 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 .

        iii. 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 .

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. Genere una solicitud de UpdateUserPoolAPI con un `LambdaConfig` parámetro que añada activadores Lambda de remitente personalizados. No puede añadir desencadenadores de este tipo en la consola de Amazon Cognito. Los desencadenadores de remitente personalizado requieren parámetros `LambdaConfig` de `KMSKeyID` y `CustomSMSSender` o `CustomEmailSender` (o ambos).

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 activador de remitente personalizado, tu AWS Lambda función puede enviar notificaciones por correo electrónico 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 de correo electrónico del grupo de usuarios.

Este desencadenador es útil en los casos en que quiera tener más control sobre la forma en que el grupo de usuarios envía mensajes de correo electrónico. Su función de Lambda puede personalizar la llamada a las operaciones de API de Amazon SES, como, por ejemplo, cuando quiera administrar varias identidades verificadas o entre Regiones de AWS. La función también podría redirigir los mensajes a otro medio de entrega o a un servicio de terceros.

Para obtener información sobre cómo configurar un desencadenador de remitente de correo electrónico personalizado, consulte Activación de desencadenadores de Lambda Remitente personalizado.

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	Event
`CustomEmailSender_SignUp`	Un usuario se registra y Amazon Cognito envía un mensaje de bienvenida.
`CustomEmailSender_Authentication`	Un usuario inicia sesión y Amazon Cognito envía un código de MFA o OTP por correo electrónico.
`CustomEmailSender_ForgotPassword`	Un usuario solicita un código para restablecer su contraseña.
`CustomEmailSender_ResendCode`	Un usuario solicita un código de sustitución de confirmación de cuenta.
`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.

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. 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

> **Note**
>
> Amazon Cognito envía `ClientMetadata` a funciones de desencadenador de correo electrónico personalizadas en eventos con los siguientes orígenes de activación:
>
> - `CustomEmailSender_ForgotPassword`

- `CustomEmailSender_SignUp`
- `CustomEmailSender_Authentication`

Amazon Cognito no envía `ClientMetadata` en eventos de desencadenador con el origen `CustomEmailSender_AccountTakeOverNotification`.

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 de Lambda debe interpretar el evento, descifrar el código y, a continuación, entregar el contenido del mensaje. Una función típica agrupa un mensaje de correo electrónico y lo dirige a un relé de SMTP de terceros.

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_ID`

El ID 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.

Cómo implementar esta función

1. Instale la última versión de NodeJS en el espacio de trabajo de desarrollador.
2. Cree un nuevo proyecto de NodeJS en su espacio de trabajo.
3. Inicialice el proyecto con `npm init -y`.
4. Cree el script para la función de Lambda: `touch index.mjs`.

5. Pegue el contenido del siguiente ejemplo en `index.mjs`.
6. Descargue la dependencia del proyecto, AWS Encryption SDK:`npm install @aws-crypto/client-node`.
7. Comprima el directorio del proyecto en un archivo: `zip -r my_deployment_package.zip .`.
8. Implemente el archivo ZIP en su función.

Esta función de ejemplo descifra el código y, para los eventos de registro, simula el envío de un mensaje de correo electrónico a la dirección de correo electrónico del usuario.

```
import { KmsKeyringNode, buildClient, CommitmentPolicy } from '@aws-crypto/client-
node';

// Configure the encryption SDK client with the KMS key from the environment variables
const { encrypt, decrypt } = buildClient(
    CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT
);

const generatorKeyId = process.env.KEY_ID;
const keyIds = [process.env.KEY_ARN];
const keyring = new KmsKeyringNode({ generatorKeyId, keyIds });

// Example function to simulate sending email.
// This example logs message details to CloudWatch Logs from your Lambda function.
// Update this function with custom logic that sends an email message to 'emailaddress'
 with body 'message'.
const sendEmail = async (emailAddress, message) => {
    // Log the destination with the email address masked.
    console.log(`Simulating email send to ${emailAddress.replace(/[^@.]/g, '*')}`);
    // Log the message with the code masked.
    console.log(`Message content: ${message.replace(/\b\d{6,8}\b/g, '********')}`);
    // Simulate API delay
    await new Promise(resolve => setTimeout(resolve, 100));
    console.log('Email sent successfully');
    return true;
};

export const handler = async (event) => {
    try {
        // Decrypt the secret code using encryption SDK
        let plainTextCode;
```

```
        if (event.request.code) {
            const { plaintext, messageHeader } = await decrypt(keyring,
 Buffer.from(event.request.code, 'base64'));
            plainTextCode = Buffer.from(plaintext).toString('utf-8');
        }

        // Handle different trigger sources
        if (event.triggerSource == 'CustomEmailSender_SignUp') {
            const emailAddress = event.request.userAttributes.email;
            const message = `Welcome! Your verification code is: ${plainTextCode}`;
            await sendEmail(emailAddress, message);
        }
        else if (event.triggerSource == 'CustomEmailSender_ResendCode') {
            // Handle resend code
        }
        else if (event.triggerSource == 'CustomEmailSender_ForgotPassword') {
            // Handle forgot password
        }
        else if (event.triggerSource == 'CustomEmailSender_UpdateUserAttribute') {
            // Handle update attribute
        }
        else if (event.triggerSource == 'CustomEmailSender_VerifyUserAttribute') {
            // Handle verify attribute
        }
        else if (event.triggerSource == 'CustomEmailSender_AdminCreateUser') {
            // Handle admin create user
        }
        else if (event.triggerSource == 'CustomEmailSender_Authentication') {
            // Handle authentication
        }
        else if (event.triggerSource ==
 'CustomEmailSender_AccountTakeOverNotification') {
            // Handle account takeover notification
        }

        return;
    } catch (error) {
        console.error('Error in custom email sender:', error);
        throw error;
    }
};
```

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.

Este desencadenador es útil en los casos en que quiera tener más control sobre la forma en que el grupo de usuarios envía mensajes SMS. Su función Lambda puede personalizar la llamada a las operaciones de la API de Amazon SNS, por ejemplo, cuando quiere gestionar varios IDs orígenes o cruces. Regiones de AWS La función también podría redirigir los mensajes a otro medio de entrega o a un servicio de terceros.

Para obtener información sobre cómo configurar un desencadenador de remitente de correo electrónico personalizado, consulte Activación de desencadenadores de Lambda Remitente personalizado.

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`	Event
`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.
`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.

TriggerSource value	Event
`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 inicia sesión y Amazon Cognito envía un código de MFA o OTP por SMS.
`CustomSMSSender_AdminCreateUser`	Crea un nuevo usuario en su grupo de usuarios y Amazon Cognito le envía una contraseña temporal.

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 la 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.

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_ID`

El ID 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.

Cómo implementar esta función

1. Instale la última versión de NodeJS en el espacio de trabajo de desarrollador.
2. Cree un nuevo proyecto de NodeJS en su espacio de trabajo.
3. Inicialice el proyecto con `npm init -y`.
4. Cree el script para la función de Lambda: `touch index.mjs`.
5. Pegue el contenido del siguiente ejemplo en `index.mjs`.
6. Descargue la dependencia del proyecto, AWS Encryption SDK:`npm install @aws-crypto/client-node`.
7. Comprima el directorio del proyecto en un archivo: `zip -r my_deployment_package.zip .`.
8. Implemente el archivo ZIP en su función.

```
import { KmsKeyringNode, buildClient, CommitmentPolicy } from '@aws-crypto/client-
node';

// Configure the encryption SDK client with the KMS key from the environment variables
const { encrypt, decrypt } = buildClient(
    CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT
);

const generatorKeyId = process.env.KEY_ID;
const keyIds = [process.env.KEY_ARN];
const keyring = new KmsKeyringNode({ generatorKeyId, keyIds });

// Example function to simulate sending SMS.
// This example logs message details to CloudWatch Logs from your Lambda function.
// Update this function with custom logic that sends an SMS message to 'phoneNumber'
 with body 'message'.
const sendSMS = async (phoneNumber, message) => {
    // Log the destination with the phone number masked.
    console.log(`Simulating SMS send to ${phoneNumber.replace(/[^+]/g, '*')}`);
    // Log the message with the code masked.
    console.log(`Message content: ${message.replace(/\b\d{6,8}\b/g, '********')}`);
    // Simulate API delay
    await new Promise(resolve => setTimeout(resolve, 100));
    console.log('SMS sent successfully');
    return true;
};
```

```
export const handler = async (event) => {
    try {
        // Decrypt the secret code using encryption SDK
        let plainTextCode;
        if (event.request.code) {
            const { plaintext, messageHeader } = await decrypt(keyring,
 Buffer.from(event.request.code, 'base64'));
            plainTextCode = Buffer.from(plaintext).toString('utf-8');
        }

        // Handle different trigger sources
        if (event.triggerSource == 'CustomSMSSender_SignUp') {
            const phoneNumber = event.request.userAttributes.phone_number;
            const message = `Welcome! Your verification code is: ${plainTextCode}`;
            await sendSMS(phoneNumber, message);
        }
        else if (event.triggerSource == 'CustomSMSSender_ResendCode') {
            // Handle resend code
        }
        else if (event.triggerSource == 'CustomSMSSender_ForgotPassword') {
            // Handle forgot password
        }
        else if (event.triggerSource == 'CustomSMSSender_UpdateUserAttribute') {
            // Handle update attribute
        }
        else if (event.triggerSource == 'CustomSMSSender_VerifyUserAttribute') {
            // Handle verify attribute
        }
        else if (event.triggerSource == 'CustomSMSSender_AdminCreateUser') {
            // Handle admin create user
        }
        return;
    } catch (error) {
        console.error('Error in custom SMS sender:', error);
        throw error;
    }
};
```

Temas

- Evaluar las capacidades de los mensajes SMS con una función de remitente de SMS personalizado

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.

Mensajería y análisis

Configure los mensajes de correo electrónico y SMS para su grupo de usuarios y configure Amazon Pinpoint Analytics para realizar un seguimiento de la participación de los usuarios.

Temas

- Configuración de correo electrónico para grupos de usuarios de Amazon Cognito
- Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito
- Uso de Amazon Pinpoint para analizar grupos de usuarios

Configuración de correo electrónico para grupos de usuarios de Amazon Cognito

En algunos eventos de su aplicación, es posible que Amazon Cognito deba enviar un correo electrónico a los 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 de 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
Contraseña olvidada	ForgotPassword, AdminResetUserPassword	Correo electrónico, SMS	code	Sí	Mensaje de verificación
Invitación	AdminCreateUser	Correo electrónico, SMS	code	Sí	Mensaje de invitación
Self-registration	SignUp, ResendConfirmationCode	Correo electrónico, SMS	código, enlace	Sí	Mensaje de verificación
Verificación de dirección de correo electrónico o número de teléfono	UpdateUserAttributes, AdminUpdateUserAttributes, GetUserAttributeVerificationCode	Correo electrónico, SMS	code	Sí	Mensaje de verificación
Multi-factor autenticación (MFA)	AdminInitiateAuth, InitiateAuth	Correo electrónico[1], SMS, aplicación de autenticación	code	Sí[2]	Mensaje de MFA
One-time autenticación por contraseña (OTP)	AdminInitiateAuth, InitiateAuth	Correo electrónico[1], SMS	code	Sí	Mensaje MFA [3]

[1] Requiere el plan de características Essentials o superior y la configuración de correo electrónico de Amazon SES.

[2] Para mensajes SMS y de correo electrónico.

[3] Solo puede personalizar la plantilla de mensaje de MFA cuando la MFA sea obligatoria u opcional en su grupo de usuarios. Cuando la MFA está inactiva, Amazon Cognito envía contraseñas de un solo uso con la plantilla predeterminada.

Amazon SES cobra por los mensajes de correo electrónico. Para obtener más información, consulte Precios de Amazon SES.

Para obtener más información sobre la MFA en los correos electrónicos, consulte MFA con mensajes SMS y correo electrónico.

Amazon Cognito podría impedir la entrega de más mensajes de correo electrónico o SMS a un único destino en un breve período de tiempo. Si cree que su grupo de usuarios está afectado, configure y revise los registros para detectar errores en la entrega de mensajes y, a continuación, póngase en contacto con el equipo de su cuenta.

Configuración predeterminada del correo electrónico

Amazon Cognito puede usar su propia configuración de correo electrónico predeterminada para gestionar las entregas de correo electrónico en su nombre. 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 se utiliza la configuración predeterminada, se usan recursos de Amazon SES administrados por 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 poder entregarse posteriormente, no se puede controlar su eliminación de la lista de supresión mientras el grupo de usuarios esté configurado para utilizar la configuración predeterminada. Una dirección de correo electrónico puede permanecer indefinidamente en la lista AWS de 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 se crea un grupo de usuarios tiene 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 varias regiones remotas. Para obtener

el mejor rendimiento y, si tiene la opción, envíe mensajes de correo electrónico con una identidad verificada de Amazon SES en la misma región que su grupo de usuarios.

Categorías de requisitos regionales para las identidades verificadas de Amazon SES

In-Region únicamente

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 correo electrónico `FROM` personalizada, Amazon Cognito utiliza una identidad verificada `no-reply@verificationemail.com` 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 característica aporta continuidad a los recursos del grupo de usuarios que pueda haber creado para cumplir los requisitos de Amazon Cognito al lanzar 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 correo electrónico `FROM` personalizada, Amazon Cognito utiliza una identidad verificada `no-reply@verificationemail.com` 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 genera cuando Amazon SES no está disponible en una región en la que Amazon Cognito sí lo está.

La política de autorización de envío de Amazon SES para su identidad verificada en la región alternativa debe confiar en la entidad principal de servicio de Amazon Cognito de la región de origen. Para obtener más información, consulte Concesión de 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 usar una dirección de correo electrónico `FROM` predeterminada, la política de autorización de envío de Amazon SES para su identidad verificada en cada región alternativa debe confiar en la entidad principal de servicio de Amazon Cognito de la región de origen. Para obtener más información, consulte Concesión de permisos para usar la configuración de correo electrónico predeterminada. Con la configuración de correo electrónico de Amazon SES de `DEVELOPER` en estas regiones, debe usar una identidad verificada en la primera región de la lista y configurarla para que confíe en la entidad principal del servicio de Amazon Cognito en la región del grupo de usuarios. Por ejemplo, en un grupo de usuarios de Medio Oriente (EAU), configure una identidad verificada en Europa (Fráncfort) para que sea de confianza para `cognito-idp.me-central-1.amazonaws.com`. En la configuración de correo electrónico predeterminada sin una dirección de correo electrónico `FROM` personalizada, Amazon Cognito utiliza una identidad verificada `no-reply@verificationemail.com` 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`.
> - Quiere usar una dirección `FROM` personalizada.
> - El grupo de usuarios envía correos electrónicos a una región alternativa.
> - El grupo de usuarios tiene una segunda[1] región 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 admitidas de 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)	In-Region únicamente	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)
Oeste de Canadá (Calgary)	Región alternativa	Canadá (centro), Oeste de EE. UU. (Norte de California)[1]
México (centro)	Región alternativa	Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón)[1]
Asia-Pacífico (Tokio)	Compatible con versiones anteriores	Asia-Pacífico (Tokio), Este de EE. UU. (Norte de Virginia),

Región del grupo de usuarios	Opción de región	Regiones admitidas de Amazon SES
		Oeste de EE. UU. (Oregón), Europa (Irlanda)
Asia-Pacífico (Hong Kong)	Región alternativa	Asia-Pacífico (Singapur), Asia-Pacífico (Tokio)[1]
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 (Malasia)	Región alternativa	Asia-Pacífico (Sídney), Asia-Pacífico (Singapur)[1]
Asia-Pacífico (Tailandia)	Región alternativa	Asia-Pacífico (Singapur), Asia-Pacífico (Mumbai)[1]
Asia-Pacífico (Mumbai)	Compatible con versiones anteriores	Asia-Pacífico (Bombay), Este de EE. UU. (Norte de Virginia) , Oeste de EE. UU. (Oregón), Europa (Irlanda)
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)

Región del grupo de usuarios	Opción de región	Regiones admitidas de Amazon SES
Asia-Pacífico (Osaka)	In-Region solo	Asia-Pacífico (Osaka)
Asia-Pacífico (Yakarta)	In-Region solo	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)	In-Region solo	Europa (París)
Europa (Fráncfort)	Compatible con versiones anteriores	Europa (Londres), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Europa (Zúrich)	Región alternativa	Europa (Fráncfort), Europa (Londres)[1]
Europa (Estocolmo)	In-Region solo	Europa (Estocolmo)
Europa (Milán)	In-Region solo	Europa (Milán)
Europa (España)	Región alternativa	Europa (París), Europa (Estocolmo)[1]
Middle East (Bahrain)	In-Region solo	Middle East (Bahrain)

Región del grupo de usuarios	Opción de región	Regiones admitidas de Amazon SES
Medio Oriente (EAU)	Región alternativa	Europa (Fráncfort), Europa (Londres)[1]
América del Sur (São Paulo)	In-Region solo	América del Sur (São Paulo)
Israel (Tel Aviv)	In-Region solo	Israel (Tel Aviv)
África (Ciudad del Cabo)	In-Region solo	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 dirección `FROM` personalizada, configure `EmailConfiguration` con un parámetro `SourceArn` con el 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 siga para concederlos dependerán de si usa la configuración predeterminada de correo electrónico o su configuración de Amazon SES.

Concesión de permisos para usar la configuración de correo electrónico predeterminada

Ejecute este paso solo si tiene el grupo de usuarios configurado en Enviar correo electrónico con Cognito o ha establecido `EmailSendingAccount` en `COGNITO_DEFAULT`.

Con la configuración de correo electrónico predeterminada, el grupo de usuarios puede enviar mensajes de correo electrónico con cualquiera de las siguientes direcciones.

- La dirección `no-reply@verificationemail.com` predeterminada.
- 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 conceden mediante una política de autorización de envío, para la dirección o dominio en 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 de la región del grupo de usuarios o de una región alternativa debe permitir a la entidad principal del servicio de Amazon Cognito enviar mensajes de correo electrónico. Para obtener más información, consulte la tabla de regiones. 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 la entidad principal del servicio global en el siguiente ejemplo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "stmnt1234567891234",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "email.cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "arn:aws:ses:us-
east-1:111122223333:identity/support@example.com",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        }
    ]
}
```

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. Este es el caso, por ejemplo, de Medio Oriente (EAU) 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 Medio Oriente (EAU), la región adicional es Europa (Londres). Debe actualizar la política de autorización de envío en ambas regiones.

Su política de autorización de envío en cada una de las regiones alternativas debe permitir a la entidad principal del servicio de Amazon Cognito en la región de suscripción del grupo de usuarios enviar mensajes de correo electrónico. Para obtener más información, consulte la tabla de regiones. Si la región está marcada como región alternativa, configure las políticas de autorización de envío con la entidad principal del servicio regional, como se muestra en el siguiente ejemplo. Sustituya el identificador *me-central-1* de región del ejemplo por el ID de región requerido, según sea necesario.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.me-central-1.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "arn:aws:ses:us-
east-1:111122223333:identity/support@example.com",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
```

```
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        }
    ]
}
```

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 REPLY-TO dirección personalizada que recibe los mensajes que los usuarios envían a su dirección de origen

- 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 FROM y dirección de REPLY-TO correo electrónico de forma predeterminada. Sin embargo, si su identidad verificada es un dominio, debe proporcionar un valor para las direcciones de correo electrónico FROM.

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 AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Seleccione el menú Métodos de autenticación, busque la configuración del correo electrónico y seleccione Editar.
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 la REPLY-TO sección Dirección de correo electrónico, introduzca la dirección de correo electrónico en la que desee recibir los mensajes que los usuarios envíen a su dirección DE ORIGEN.

7. Seleccione Save changes (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.

Amazon Cognito puede enviar mensajes SMS con una de las siguientes configuraciones:

- Amazon Simple Notification Service (Amazon SNS): Amazon Cognito llama a Amazon SNS, que enruta los mensajes SMS a. AWS End User Messaging SMS
- AWS End User Messaging SMS directo: Amazon Cognito llama AWS End User Messaging SMS directamente, sin Amazon SNS como intermediario. Puede utilizar esta opción en la mayoría de los Regiones de AWS lugares donde Amazon Cognito esté disponible. Para conocer las excepciones, consulte AWS End User Messaging SMS ruta directa: configuración regional.

Un grupo de usuarios utiliza una configuración de SMS a la vez. Las configuraciones Amazon SNS y AWS End User Messaging SMS Direct se excluyen mutuamente. Al establecer una configuración, Amazon Cognito borra la otra. Esto se aplica a las operaciones CreateUserPooly a UpdateUserPoollas mismas. SetUserPoolMfaConfig

Amazon SNS SMS está disponible en la Regiones de AWS lista de. Configuración regional de la ruta de Amazon SNS Para enviar mensajes SMS desde un grupo de usuarios de cualquier otro grupo Región de AWS, utilícelo AWS End User Messaging SMS directamente.

Si envía un mensaje de texto a través de Amazon Cognito por primera vez, lo AWS End User Messaging SMS coloca en un entorno sandbox. En el entorno aislado, puede probar los mensajes de texto SMS de sus aplicaciones. En el entorno limitado, puede enviar mensajes a un número limitado de números de teléfono de destino verificados si tiene una identidad de origen, o simular el envío de mensajes si no la tiene.

> **Note**
>
> En noviembre de 2024, AWS sustituyó la mensajería SMS de Amazon SNS por. AWS End User Messaging SMS Cuando utiliza la ruta de configuración de Amazon SNS, los grupos de usuarios inician los mensajes SMS con la operación Amazon SNS Publish, que es una transferencia a la que se transfiere. AWS End User Messaging SMS Para esta ruta, debe configurar los permisos para. `sns:Publish` Cuando utiliza la ruta de AWS End User Messaging SMS configuración directa, Amazon Cognito llama a la AWS End User Messaging SMS `SendTextMessage` operación y usted debe configurar los permisos para. `sms-voice:SendTextMessage`

AWS End User Messaging SMS cargos por mensajes de texto SMS. Para obtener más información, consulte Precios de AWS End User Messaging SMS.

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, AdminResetUserPassword	Correo electrónico, SMS	code	Sí	Mensaje de verificación

Actividad	Operación de la API	Opciones de entrega	Opciones de formato	Personalizable	Plantilla de mensaje
Invitación	AdminCreateUser	Correo electrónico, SMS	code	Sí	Mensaje de invitación
Self-registration	SignUp, ResendConfirmationCode	Correo electrónico, SMS	código, enlace	Sí	Mensaje de verificación
Verificación de dirección de correo electrónico o número de teléfono	UpdateUserAttributes, AdminUpdateUserAttributes, GetUserAttributeVerificationCode	Correo electrónico, SMS	code	Sí	Mensaje de verificación
Multi-factor autenticación (MFA)	AdminInitiateAuth, InitiateAuth	Correo electrónico[1], SMS, aplicación de autenticación	code	Sí[2]	Mensaje MFA
One-time autenticación por contraseña (OTP)	AdminInitiateAuth, InitiateAuth	Correo electrónico[1], SMS	code	Sí	Mensaje MFA [3]

[1] Requiere el plan de características Essentials o superior y la configuración de correo electrónico de Amazon SES.

[2] Para mensajes SMS y de correo electrónico.

[3] Solo puede personalizar la plantilla de mensaje de MFA cuando la MFA sea obligatoria u opcional en su grupo de usuarios. Cuando la MFA está inactiva, Amazon Cognito envía contraseñas de un solo uso con la plantilla predeterminada.

AWS End User Messaging SMS cargos por mensajes SMS. Para obtener más información, consulte Precios de AWS End User Messaging SMS.

Para obtener más información sobre MFA, consulte MFA con mensajes SMS y correo electrónico.

Amazon Cognito podría impedir la entrega de más mensajes de correo electrónico o SMS a un único destino en un breve período de tiempo. Si cree que su grupo de usuarios está afectado, configure y revise los registros para detectar errores en la entrega de mensajes y, a continuación, póngase en contacto con el equipo de su cuenta.

Elegir entre Amazon SNS y AWS End User Messaging SMS

Los grupos de usuarios de Amazon Cognito pueden enviar mensajes SMS a través de dos rutas: Amazon AWS End User Messaging SMS SNS o directamente. Para los grupos de usuarios nuevos, utilice la ruta AWS End User Messaging SMS directa. Amazon SNS SMS solo está disponible en un conjunto fijo de. Regiones de AWS La AWS End User Messaging SMS ruta directa admite regiones adicionales y expone características que la ruta Amazon SNS no incluye. Ambas rutas envían mensajes a través de ellas. AWS End User Messaging SMS

En la siguiente tabla se comparan las dos rutas de configuración de SMS.

Funcionalidad	Ruta de Amazon SNS	AWS End User Messaging SMS ruta directa
Disponibilidad por región	Disponible en la Regiones de AWS tabla de regiones de rutas de Amazon SNS	Disponible en la mayoría de los Regiones de AWS lugares donde Amazon Cognito está disponible. Las regiones en las que solo está disponible la ruta Amazon SNS aparecen en la lista. AWS End User Messaging SMS ruta directa: configuración regional
Integración	Amazon Cognito denomina a la operación Amazon SNS	Amazon Cognito llama directamente a la operación

Funcionalidad	Ruta de Amazon SNS	AWS End User Messaging SMS ruta directa
	Publish, que pasa a AWS End User Messaging SMS	AWS End User Messaging SMS SendTextMessage
Permiso de IAM	`sns:Publish`	`sms-voice:SendTextMessage`
Conjuntos de configuraciones	No se puede configurar a través de Amazon Cognito	Establezca el `ConfigurationSetName` campo para el registro de eventos y los destinos
Identidad de origen	No se puede configurar a través de Amazon Cognito	Configure el campo `OriginationIdentity`
Cross-Region enviando	Establece el `SnsRegion` campo	No compatible
Registro de DLT en India	No se puede configurar a través de Amazon Cognito	Configure los campos `InEntityId` y `InTemplateId`
Protección contra el fraude (Proteger)	Account-default se aplica la configuración de protección	Account-default se aplica. También puede aplicar una configuración de protección a un caso de uso específico mediante un conjunto de configuraciones.

> **Note**
>
> La configuración de AWS End User Messaging SMS protección predeterminada se aplica a todos los mensajes, incluidos los de Amazon Cognito. Para aplicar una configuración de protección a un caso de uso específico, vincúlela a un conjunto de configuraciones y utilice

la AWS End User Messaging SMS ruta directa. Para obtener más información, consulte Protección contra el fraude en AWS End User Messaging SMS.

Configuración de SMS en la API de grupos de usuarios de Amazon Cognito

El SmsConfigurationTypeobjeto de la API de grupos de usuarios de Amazon Cognito contiene la configuración de SMS de su grupo de usuarios. Este objeto tiene las siguientes rutas de configuración:

Ruta de Amazon SNS

`SnsCallerArn`Establézcalo en el ARN de un rol de IAM que conceda permiso a Amazon Cognito para realizar llamadas. `sns:Publish` Si lo desea, defina y. `SnsRegion` `ExternalId` El `SnsCallerArn` parámetro es opcional cuando se utiliza la ruta AWS End User Messaging SMS directa.

AWS End User Messaging SMS ruta directa

Establece el tipo de `EumsSms` objeto`EumsSmsConfigurationType`. Este objeto tiene los siguientes miembros:

- `CallerArn`(Obligatorio): el ARN de un rol de IAM que otorga permiso a Amazon Cognito para realizar llamadas. `sms-voice:SendTextMessage`
- `ExternalId`(Opcional): el ID externo de la política de confianza de roles de IAM. Este valor evita que los diputados se confundan.
- `OriginationIdentity`(Opcional): el número de teléfono, el grupo o el identificador del remitente que se va a utilizar para el envío.
- `ConfigurationSetName`(Opcional): el conjunto de AWS End User Messaging SMS configuraciones que se va a utilizar.
- `InEntityId`(Opcional): el ID de registro de la entidad del registro de DLT de India. Necesario para enviar a destinatarios en India.
- `InTemplateId`(Opcional): el ID de registro de la plantilla del registro de DLT de India. Necesario para enviar a destinatarios en India.
- `Region`(Opcional): uno Región de AWS de tus AWS End User Messaging SMS recursos. Para obtener más información, consulte Elija el icono Región de AWS para mensajes SMS.

Para obtener más información, consulte la SmsConfigurationTypereferencia de la API de grupos de usuarios de Amazon Cognito.

Prácticas recomendadas

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 SMS-message-related las normativas de los países en los que puedan residir sus usuarios y mantener actualizada la configuración de sus mensajes SMS. Para obtener más información, consulte Capacidades y limitaciones de los SMS por país en la Guía del usuario de AWS End User Messaging SMS .

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.

Consulte los siguientes enlaces para obtener información sobre cómo proteger su arquitectura de entrega de mensajes SMS.

- Reducción de los riesgos de fraude en el registro de usuarios y del fraude de tráfico artificial de SMS con los grupos de usuarios de Amazon Cognito
- Cómo defenderse del tráfico masivo de SMS: nuevas AWS funciones que ayudan a combatir el tráfico inflado artificialmente

Configuración de mensajes SMS por primera vez en grupos de usuarios de Amazon Cognito

Amazon Cognito utiliza Amazon SNS (e AWS End User Messaging SMS indirectamente) AWS End User Messaging SMS o directamente para enviar mensajes SMS desde sus 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 configure los mensajes de texto SMS en una región concreta Región de AWS, estará AWS End User Messaging SMS Cuenta de AWS en el entorno limitado de SMS de esa región. AWS End User Messaging SMS utiliza el entorno de pruebas para evitar el fraude y el abuso y para cumplir con los requisitos de

conformidad. Cuando Cuenta de AWS está en un entorno de pruebas, AWS End User Messaging SMS impone algunas restricciones. Por ejemplo, puede enviar mensajes de texto a un máximo de 10 números de destino verificados si tiene una identidad de origen, o puede simular el envío de mensajes sin una identidad de origen. Mientras estés en Cuenta de AWS la zona de pruebas, no envíes mensajes SMS durante el proceso de producción. Cuando se encuentra en el entorno de pruebas, Amazon Cognito no puede enviar mensajes a los números de teléfono de sus usuarios.

Temas

- Prepare una función de IAM que Amazon Cognito pueda utilizar para enviar mensajes SMS con AWS End User Messaging SMS
- Para configurar AWS End User Messaging SMS SMS directo para un grupo de usuarios
- Elija el icono Región de AWS para mensajes SMS
- Obtener una identidad de origen para enviar mensajes SMS a números de teléfono de EE. UU.
- Confirmar que se encuentra en el entorno de pruebas de SMS
- Traslado de la cuenta fuera del entorno de pruebas
- Utilice números de simulador o números de teléfono verificados con AWS End User Messaging SMS
- Completar la configuración del grupo de usuarios en Amazon Cognito

Prepare una función de IAM que Amazon Cognito pueda utilizar para enviar mensajes SMS con AWS End User Messaging SMS

Cuando envía un mensaje SMS desde su grupo de usuarios, Amazon Cognito asume un rol de IAM en su cuenta. Los permisos que requiere el rol dependen de la ruta de configuración de SMS que elija:

- Ruta de Amazon SNS: el rol debe conceder el `sns:Publish` permiso. Amazon Cognito utiliza este permiso para enviar mensajes SMS a través de Amazon SNS.
- AWS End User Messaging SMS ruta directa: el rol debe conceder el `sms-voice:SendTextMessage` permiso. Amazon Cognito utiliza este permiso para llamar AWS End User Messaging SMS directamente.

En la consola de Amazon Cognito, puede configurar una selección de roles de IAM desde el menú Métodos de autenticación de su grupo de usuarios, en SMS, o realizar esta selección durante 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 asumir el rol si cumple las siguientes condiciones:

- La operación de asumir el rol se realiza en nombre del grupo de usuarios de la condición `aws:SourceArn`.
- La operación de asumir el rol se realiza en nombre de un grupo de usuarios de la Cuenta de AWS establecida mediante la condición `aws:SourceAccount`.
- La operación de asumir el rol incluye el ID externo en la condición `sts:externalId`.

Estas condiciones de la política de confianza se aplican tanto a la ruta Amazon SNS como a la ruta AWS End User Messaging SMS directa.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-idp.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:cognito-idp:us-
west-2:111122223333:userpool/us-west-2_EXAMPLE"
                }
            }
        }
    ]
}
```

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 Consola de administración de AWS o con una solicitud de DescribeUserPoolAPI.

Para enviar mensajes SMS para la autenticación multifactor, la política de confianza del rol de IAM debe tener una condición `sts:ExternalId`. El valor de esta condición debe coincidir con la `ExternalId` propiedad de su grupo SmsConfigurationde usuarios. Cuando crea un rol de IAM durante el proceso de creación del grupo de usuarios en la consola de Amazon Cognito, Amazon Cognito configura en su lugar el ID externo en los ajustes del rol y del grupo de usuarios. Este no es el caso cuando usa un rol de IAM que ya existe.

Debe actualizar el `ExternalId` parámetro del grupo de usuarios en una solicitud de UpdateUserPoolAPI y actualizar la política de confianza de roles de IAM con una `sts:externalId` condición con el mismo valor. Para obtener información sobre cómo usar la API para actualizar un grupo de usuarios de forma que se conserve la configuración original, consulte Actualización de la configuración del grupo de usuarios y del cliente de aplicación.

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 .

Para configurar AWS End User Messaging SMS SMS directo para un grupo de usuarios

Utilice el siguiente procedimiento para configurar su grupo de usuarios para que envíen mensajes SMS directamente a través de él AWS End User Messaging SMS. Esta configuración utiliza el `EumsSms` objeto en SmsConfigurationType.

Para configurar AWS End User Messaging SMS mensajería SMS directa

1. Cree o identifique una AWS End User Messaging SMS identidad de origen en la Región de AWS que desee enviar. La identidad de origen puede ser un número de teléfono, un identificador de remitente o un grupo. También puede crear un conjunto de AWS End User Messaging SMS configuraciones para aplicar las reglas de envío. Para obtener instrucciones sobre cómo solicitar un número de teléfono, consulte Solicitar un número de teléfono en la Guía del AWS End User Messaging SMS usuario.
2. Cree un rol de IAM al que Amazon Cognito pueda asumir que lo llama. AWS End User Messaging SMS La política de confianza para este rol es la misma que la ruta de Amazon SNS: debe incluir `aws:SourceArnaws:SourceAccount`, y `sts:ExternalId` condiciones. Para obtener información detallada sobre la política de confianza, consultePrepare una función

de IAM que Amazon Cognito pueda utilizar para enviar mensajes SMS con AWS End User Messaging SMS.

3. Adjunte la siguiente política de identidad (permisos) al rol. Esta política otorga a Amazon Cognito permiso para utilizar la identidad `sms-voice:SendTextMessage` de origen desde la que envía el grupo de usuarios. Limite el `Resource` elemento al ARN de su número de teléfono, grupo de teléfonos o ID de remitente como práctica de privilegio mínimo. Sustituya el ARN de ejemplo por el ARN de su identidad de origen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sms-voice:SendTextMessage"
            ],
            "Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-a1b2c3d4e5f6EXAMPLE"
        }
    ]
}
```

El ARN de identidad de origen utiliza uno de los siguientes formatos, según el tipo de identidad:

- Número de teléfono: `arn:`*`partition`*`:sms-voice:`*`region`*`:`*`account`*`:phone-number/`*`phoneNumberId`*
- Grupo de teléfonos — `arn:`*`partition`*`:sms-voice:`*`region`*`:`*`account`*`:pool/`*`poolId`*
- ID del remitente — `arn:`*`partition`*`:sms-voice:`*`region`*`:`*`account`*`:sender-id/`*`senderId`*`/`*`isoCountryCode`*

> **Important**
>
> Cuando su política limite el `Resource` elemento a un ARN de identidad de origen, establezca el `OriginationIdentity` campo en ese mismo ARN, por ejemplo,. `arn:aws:sms-voice:us-east-1:123456789012:phone-number/phone-a1b2c3d4e5f6EXAMPLE` Un número de teléfono con un E.164 formato (por

ejemplo+12065550100) no coincide con el ámbito de la política, por lo que la solicitud no está autorizada y Amazon Cognito no puede enviar mensajes SMS.

4. Si envía mensajes SMS a destinatarios de la India, registre un identificador de entidad de DLT y un identificador de plantilla en un portal de tecnología de contabilidad distribuida (DLT). Proporcione el ID de entidad y el ID de plantilla tal `InEntityId` y como aparecen `InTemplateId` en la configuración. `EumsSms` Para obtener más información sobre el registro de DLT en India, consulta Requisitos especiales para el envío de mensajes SMS a destinatarios de la India en la Guía del AWS End User Messaging SMS usuario.
5. Pruebe la configuración llamando directamente a la operación de la AWS End User Messaging SMS SendTextMessageAPI con la identidad de origen y el rol de IAM que haya configurado. Confirme que el mensaje de prueba se ha entregado antes de asignar la configuración a su grupo de usuarios.
6. Establezca el `EumsSms` objeto en su grupo de usuarios. Puede establecer esta configuración en una SetUserPoolMfaConfigsolicitud CreateUserPoolUpdateUserPool, o. El parámetro `CallerArn` es obligatorio. El siguiente ejemplo muestra la `EumsSms` configuración mínima:

```
"EumsSms": {
    "CallerArn": "arn:aws:iam::123456789012:role/service-role/CognitoEumsSmsRole",
    "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para enviar desde una identidad de origen específica o utilizar un conjunto de configuraciones, incluya los `ConfigurationSetName` parámetros `OriginationIdentity` y.

Permisos de IAM para AWS End User Messaging SMS SMS directos

La función de IAM `CallerArn` requiere una política de confianza que permita que el director del `cognito-idp.amazonaws.com` servicio la asuma. Utilice la misma estructura de políticas de confianza que la ruta de Amazon SNS (consultePrepare una función de IAM que Amazon Cognito pueda utilizar para enviar mensajes SMS con AWS End User Messaging SMS). El rol también requiere una política de permisos que otorgue un `sms-voice:SendTextMessage` ámbito a su ARN de identidad de origen (consulte el procedimiento de configuración en). Para configurar AWS End User Messaging SMS SMS directo para un grupo de usuarios

Elija el icono Región de AWS para mensajes SMS

> **Note**
>
> Los mensajes SMS entrantes ahora AWS se gestionan en AWS End User Messaging SMS.

Configuración regional de la ruta 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.

Elija una región para los recursos SMS en el paso Configurar entrega de mensajes del nuevo asistente del grupo de usuarios. También puede seleccionar Editar en SMS, en el menú Métodos de autenticación 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 (hasta AWS End User Messaging SMS) establece la cuota de gasto para todas las cuentas nuevas en 1 dólar (USD) al mes. Es posible que haya aumentado el límite de gasto en una Región de AWS que utilice 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 Pasar del entorno limitado de AWS End User Messaging SMS MMS y voz a la producción en la Guía del AWS End User Messaging SMS usuario.

Puede enviar mensajes SMS a cualquier región de Amazon Cognito de la siguiente tabla con los AWS End User Messaging SMS recursos de la región de mensajes SMS correspondiente.

Región de Amazon Cognito	Región de mensajes SMS
Este de EE. UU. (Ohio)	EE. UU. Este (Ohio), EE. UU. Este (Norte de Virginia)
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)
Canadá (centro)	Canadá (centro), este de EE. UU. (Norte de Virginia)
Oeste de Canadá (Calgary)	Oeste de Canadá (Calgary)
México (centro)	México (centro)
Europa (Fráncfort)	Europa (Fráncfort), Europa (Irlanda)
Europa (Londres)	Europa (Londres), Europa (Irlanda)
Europa (Irlanda)	Europa (Irlanda)
Europa (París)	Europa (París)
Europa (Estocolmo)	Europa (Estocolmo)
Europa (Milán)	Europa (Milán)
Europa (España)	Europa (España)
Europa (Zúrich)	Europa (Zúrich)
Asia-Pacífico (Malasia)	Asia-Pacífico (Singapur)
Asia-Pacífico (Tailandia)	Asia-Pacífico (Mumbai)

Región de Amazon Cognito	Región de mensajes SMS
Asia-Pacífico (Mumbai)	Asia-Pacífico (Bombay), Asia-Pacífico (Singapur)
Asia-Pacífico (Hyderabad)	Asia-Pacífico (Hyderabad)
Asia-Pacífico (Hong Kong)	Asia-Pacífico (Singapur)
Asia-Pacífico (Seúl)	Asia-Pacífico (Tokio)
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)
Asia-Pacífico (Melbourne)	Asia-Pacífico (Melbourne)
Middle East (Bahrain)	Middle East (Bahrain)
Medio Oriente (EAU)	Medio Oriente (EAU)
América del Sur (São Paulo)	América del Sur (São Paulo)
Israel (Tel Aviv)	Israel (Tel Aviv)
África (Ciudad del Cabo)	África (Ciudad del Cabo)

AWS End User Messaging SMS ruta directa: configuración regional

AWS End User Messaging SMS está disponible en su propio conjunto de Regiones de AWS, que puede diferir del de las regiones en las que están disponibles los grupos de usuarios de Amazon SNS o Amazon Cognito. Para ver la lista actual, consulte las regiones y países compatibles en la Guía del AWS End User Messaging SMS usuario.

El `Region` campo del `EumsSms` objeto SmsConfigurationTypeestablece el Región de AWS lugar al que llama Amazon Cognito. AWS End User Messaging SMS AWS End User Messaging SMS debe estar disponible en la región de su grupo de usuarios. Omita `Region` los AWS End User Messaging SMS usos de Amazon Cognito en la región de su grupo de usuarios o `Region` establézcalos en la región de su grupo de usuarios de forma explícita. Amazon Cognito rechaza cualquier otro valor.

> **Note**
>
> Si lo establece `Region` en un valor no admitido, las `SetUserPoolMfaConfig` solicitudes `CreateUserPool` `UpdateUserPool`, y fallarán con un. `InvalidParameterException`

> **Note**
>
> La ruta AWS End User Messaging SMS directa no está disponible en Asia Pacífico (Tailandia), Asia Pacífico (Hong Kong) y Asia Pacífico (Malasia Regiones de AWS). Para enviar mensajes SMS desde un grupo de usuarios de una de estas regiones, utilice la ruta Amazon SNS en su lugar. Para obtener más información, consulte Configuración regional de la ruta de Amazon SNS.

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.

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 de teléfono en la Guía del usuario de AWS End User Messaging SMS .

Si tiene más de una identidad de origen en la misma Región de AWS, AWS End User Messaging SMS elija 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 Preguntas frecuentes de AWS End User Messaging SMS.

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 menú Métodos de autenticación.
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.

   Configure Servicio de AWS las dependencias para completar la configuración de sus mensajes SMS

   Si no ves este mensaje, significa que alguien ya ha configurado los mensajes SMS en tu cuenta. Vaya a Completar la configuración del grupo de usuarios en Amazon Cognito.
6. Elija el enlace Amazon SNS en Pasar al entorno de producción de Amazon SNS. 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 tu entorno aislado y Región de AWS, de la siguiente manera:

   ```
   This account is in the SMS sandbox in US East (N. Virginia).
   ```

Traslado de la cuenta fuera del entorno de pruebas

Para utilizar su aplicación en producción, quite la cuenta del entorno aislado de SMS y entre en producción. Una vez que haya configurado una identidad de origen Región de AWS que contenga los AWS End User Messaging SMS recursos que desea que utilice Amazon Cognito, podrá verificar los números de teléfono de EE. UU. mientras permanece en el Cuenta de AWS entorno limitado de SMS. Cuando su entorno está en producción, no necesita verificar los números de teléfono de los usuarios antes de enviarles mensajes SMS.

Puede crear una solicitud para salir del sandbox desde la AWS End User Messaging SMS consola o desde la consola de Amazon SNS. Para obtener instrucciones detalladas, consulte Traslado desde el entorno de pruebas SMS en la Guía del usuario de AWS End User Messaging SMS .

Utilice números de simulador o números de teléfono verificados con AWS End User Messaging SMS

Si ha quitado la cuenta del entorno aislado de SMS, omita este paso.

Si está en un entorno de pruebas pero ha configurado un número de origen, puede enviar mensajes a números de destino verificados. Para configurar destinos verificados, consulte Cómo agregar un número de teléfono de destino verificado en la Guía del usuario de AWS End User Messaging SMS .

También puede enviar mensajes con remitentes y destinos simulados. Los mensajes del simulador producen registros, pero no se envían a través de la red del operador. En el Menú de atajos, seleccione Probar el envío de SMS con el simulador de SMS. Para obtener más información, consulte Números de teléfono de simuladores en la Guía del usuario de AWS End User Messaging SMS .

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 por SMS a su grupo de usuarios, Amazon Cognito enviará un mensaje de prueba a un número de teléfono interno para comprobar que la configuración funciona. AWS End User Messaging SMS cobra por cada mensaje SMS de prueba.

Uso de Amazon Pinpoint para analizar grupos de usuarios

> **Note**
>
> Aviso de fin de soporte: el 30 de octubre de 2026, AWS finalizará el soporte para Amazon Pinpoint. Después del 30 de octubre de 2026, ya no podrá acceder a la consola de Amazon Pinpoint ni a los recursos de Amazon Pinpoint (puntos de conexión, segmentos, campañas, recorridos y análisis). Para obtener más información, consulte Fin de soporte de Amazon Pinpoint. Nota: en lo APIs que respecta a los SMS, este cambio no afecta a los mensajes de voz, a las notificaciones push móviles, a las OTP y a la validación de números de teléfono, y son compatibles con la mensajería para el usuario AWS final.

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. Con el soporte analítico de Amazon Pinpoint en los grupos de usuarios de Amazon Cognito, puede realizar un seguimiento de las inscripciones, los inicios de sesión, las autenticaciones fallidas, los usuarios activos diarios () y los usuarios activos mensuales DAUs () en la consola de Amazon Pinpoint. MAUs 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 Compartir datos de atributos del usuario con Amazon Pinpoint en la configuración de Análisis del cliente de aplicación en el menú Clientes de aplicación, en 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. El director de IAM que añade la configuración de análisis debe tener permisos. CreateServiceLinkedRole La función vinculada 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 Región de AWS las asignaciones 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)
- Asia-Pacífico (Malasia)

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

Región del grupo de usuarios de Amazon Cognito	Región del proyecto de Amazon Pinpoint
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
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 todos, Regiones de AWS excepto los de la tabla anterior, Amazon Cognito solo puede usar 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 (Consola de administración de AWS)

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 el menú Clientes de aplicación y seleccione el cliente de aplicación que desee actualizar.
4. En la pestaña Análisis, en Análisis de Pinpoint, seleccione Activar.
5. Elija un valor de Pinpoint Region (Región de Pinpoint).
6. 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.

7. 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.

8. Seleccione Save changes (Guardar cambios).

Especificación de la configuración de análisis (AWS CLI y AWS la API) de Amazon Pinpoint

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`
- AWS API: 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`
- AWS API: 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 AWS SDK mediante una aplicación con una interfaz de usuario (UI) personalizada. No se pueden pasar metadatos analíticos para usuarios federados o locales que inician sesión a través del inicio de sesión administrado. 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 JavaScript ejemplo, consulte ¿Por qué los análisis de mi grupo de usuarios de Amazon Cognito no aparecen en mi panel de Amazon Pinpoint? en el Centro de AWS conocimiento.

Protección de grupos de usuarios con protección contra amenazas

Probablemente desee proteger su aplicación contra las intrusiones en la red, la adivinación de contraseñas, la suplantación de identidad de usuarios y el registro e inicio de sesión malintencionados. La configuración de las características de seguridad de los grupos de usuarios de Amazon Cognito puede ser un componente clave en la arquitectura de seguridad. La seguridad de su aplicación es responsabilidad del cliente («Seguridad en la nube»), tal como se describe en el modelo de responsabilidad AWS compartida. Las herramientas de este capítulo contribuyen a que el diseño de seguridad de su aplicación se ajuste a estos objetivos.

Para obtener una guía de seguridad a nivel de cuenta y de todo el servicio (el modelo de responsabilidad compartida, la protección de datos, el cumplimiento y la seguridad de la infraestructura), consulte. Seguridad en Amazon Cognito

Al configurar el grupo de usuarios debe tomar una decisión importante sobre si permitirá o no el registro y el inicio de sesión públicos. Algunas opciones de grupos de usuarios, como los clientes confidenciales, la creación y confirmación administrativa de usuarios y los grupos de usuarios sin un dominio, están menos expuestas a los ataques a través de Internet. Sin embargo, suele ocurrir que los clientes públicos acepten el registro de cualquier usuario de Internet y envíen todas las operaciones directamente a su grupo de usuarios. En todas las configuraciones, pero especialmente en el caso de las configuraciones públicas, le recomendamos que planifique e implemente el grupo de usuarios teniendo siempre en cuenta las características de seguridad. La falta de seguridad también puede afectar a tu AWS factura si fuentes no deseadas crean nuevos usuarios activos o intentan aprovecharse de los usuarios existentes.

La protección contra amenazas y MFA se aplica a los usuarios locales. Third-party IdPs son responsables de la postura de seguridad de los usuarios federados.

Características de seguridad de los grupos de usuarios

Multi-factor autenticación (MFA)

Solicita un código que tu grupo de usuarios envíe por correo electrónico (con el plan de funciones Essentials o Plus) o por mensaje SMS, o desde una aplicación de autenticación, para confirmar el inicio de sesión del grupo de usuarios.

Protección contra amenazas

Supervise el inicio de sesión para detectar indicadores de riesgo y aplique la MFA o bloquee el inicio de sesión. Añada notificaciones y ámbitos personalizados para acceder a los tokens. Envíe los códigos de la MFA por correo electrónico.

AWS WAF ACL web

Inspeccione el tráfico entrante a los puntos de conexión del grupo de usuarios y a la API de autenticación para detectar actividad no deseada en las capas de red y de aplicaciones.

Sensible a mayúsculas y minúsculas

No permita la creación de usuarios cuya dirección de correo electrónico o nombre de usuario preferido sean idénticos a los de otro usuario salvo en el uso de mayúsculas y minúsculas.

Protección contra eliminación

Evite que los sistemas automatizados eliminen accidentalmente los grupos de usuarios. Exija una confirmación adicional de la eliminación del grupo de usuarios en la Consola de administración de AWS.

Errores de existencia del usuario

Protéjase de la divulgación de los nombres de usuario y alias contenidos en el grupo de usuarios. Devuelva un error genérico en respuesta a una autenticación fallida, independientemente de que el nombre de usuario sea válido o no.

Temas

- Prácticas recomendadas de seguridad de los grupos de usuarios de Amazon Cognito
- Adición de MFA a un grupo de usuarios.
- Niveles de autenticación con afirmaciones de ACR y AMR
- Uso de dispositivos de usuario en el grupos de usuarios
- Seguridad avanzada con protección contra amenazas
- Asocie un AWS WAF ACL web a un grupo de usuarios
- Sensibilidad de mayúsculas y minúsculas en el grupo de usuarios
- Protección de eliminación de grupo de usuarios
- Administración de las respuestas de error de existencia de usuarios

Prácticas recomendadas de seguridad de los grupos de usuarios de Amazon Cognito

En esta página, se describen las prácticas recomendadas de seguridad que puede implementar cuando desee protegerse contra las amenazas más comunes. La configuración que elija depende del caso de uso de cada aplicación. Le recomendamos que, como mínimo, aplique los privilegios mínimos a las operaciones administrativas y tome medidas para proteger los secretos de las aplicaciones y los usuarios. Otro paso avanzado pero eficaz que puede tomar es configurar y aplicar las ACL AWS WAF web a sus grupos de usuarios.

Protección de un grupo de usuarios en el nivel de red

AWS WAF Las ACL web pueden proteger el rendimiento y el coste de los mecanismos de autenticación que cree con Amazon Cognito. Con las ACL web, puede implementar barreras de protección frente a la API y a las solicitudes de inicio de sesión administrado. Las ACL web crean filtros de capa de red y aplicación que pueden reducir el tráfico o requerir un CAPTCHA, en función de las reglas que usted diseñe. Las solicitudes no se transfieren a sus recursos de Amazon Cognito hasta que cumplen con los requisitos de las reglas de ACL web. Para obtener más información, consulte ACL web de AWS WAF.

Protección contra el abuso de los mensajes SMS

Al permitir el registro público en su grupo de usuarios, puede configurar la verificación de la cuenta con los códigos que Amazon Cognito envía en mensajes de texto SMS. Los mensajes SMS se pueden asociar a actividades no deseadas y aumentar su AWS factura. Configure su infraestructura para que sea resistente al envío de mensajes SMS en circunstancias de fraude. Para obtener más información, consulta las siguientes publicaciones de los AWS blogs.

- Reducción de los riesgos de fraude en el registro de usuarios y del fraude de tráfico artificial de SMS con los grupos de usuarios de Amazon Cognito
- Cómo defenderse del tráfico masivo de SMS: nuevas AWS funciones que ayudan a combatir el tráfico inflado artificialmente

Funcionamiento de la autenticación pública

Los grupos de usuarios de Amazon Cognito tienen características de administración de acceso e identidad de los clientes (CIAM) que admiten casos de uso en los que los miembros del público

en general pueden registrarse para obtener una cuenta de usuario y acceder a sus aplicaciones. Cuando un grupo de usuarios permite el registro en régimen de autoservicio, está abierto a las solicitudes de cuentas de usuario procedentes de la Internet pública. Self-service las solicitudes provienen de operaciones de la API, como «y» InitiateAuth, SignUpy de la interacción del usuario con el inicio de sesión gestionado. Puede configurar grupos de usuarios para mitigar los abusos que puedan derivarse de las solicitudes públicas o deshabilitar por completo las operaciones de autenticación pública.

Los siguientes ajustes son algunas de las formas en que puede administrar las solicitudes de autenticación públicas e internas en sus grupos de usuarios y clientes de aplicación.

Ejemplos de configuraciones de grupos de usuarios que afectan al acceso a grupos de usuarios públicos

Opción	Opciones disponibles	Ubicación de la configuración	Efecto en la autenticación pública	Configuración de la consola	Parámetro y operación API
Self-serviceregistrarse	Permite a los usuarios registrarse para obtener una cuenta o crear cuentas de usuario como administradores.	Grupo de usuarios	Evita el registro público	Sign-up—registrarse Self-service	CreateUserPool, UpdateUserPool `AdminCreateUserConfig – AllowAdminCreateUserOnly`
Confirmación del administrador	Envía los códigos de confirmación a los nuevos usuarios o solicita a los administradores que los confirmen.	Grupo de usuarios	Impide la confirmación del registro sin la intervención del administrador	Sign-up—Cognito-assisted verificación y confirmación	CreateUserPool, UpdateUserPool `AccountRecoverySettings`

Opción	Opciones disponibles	Ubicación de la configuración	Efecto en la autenticación pública	Configuración de la consola	Parámetro y operación API
					`_admin_only`
Divulgación del usuario	Envía mensajes de “user not found” al iniciar sesión y restablecer la contraseña o evita que se divulguen.	Cliente de aplicación	Protección contra la adivinación del nombre de inicio de sesión, la dirección de correo electrónico o los números de teléfono	Clientes de aplicación - Evitar errores de existencia de usuarios	CreateUserPoolClient, UpdateUserPoolClient `PreventUserExistenceErrors`
Secreto del cliente	Exige o no un hash secreto al registrarse, iniciar sesión o restablecer contraseña	Cliente de aplicación	Protege frente a las solicitudes de autenticación de fuentes no autorizadas	Clientes de aplicación - Secreto de cliente	CreateUserPoolClient `GenerateSecret`
ACL web	Habilita o no un firewall de red para las solicitudes de autenticación	Grupo de usuarios	Limita o impide el acceso en función de las características de las solicitudes definidas por el administrador y las reglas de direcciones IP	AWS WAF – Configuración de WAF	AssociateWebACL `ResourceArn`

Opción	Opciones disponibles	Ubicación de la configuración	Efecto en la autenticación pública	Configuración de la consola	Parámetro y operación API
IdP externo	Permiten que los usuarios inicien sesión en un directorio de terceros IdPs, en el grupo de usuarios o en ambos	Cliente de aplicación	Excluye a los usuarios locales o los usuarios federados del registro y el inicio de sesión.	Clientes de aplicación - Proveedores de identidad	CreateUserPoolClient, UpdateUserPoolClient `SupportedIdentityProviders`
Servidor de autorización	Aloja o no páginas web públicas para la autenticación	Grupo de usuarios	Desactive las páginas web públicas y permita únicamente SDK-based la autenticación	Dominio	CreateUserPoolDomain La creación de cualquier dominio de grupo de usuarios hace que las páginas web públicas estén disponibles.

Opción	Opciones disponibles	Ubicación de la configuración	Efecto en la autenticación pública	Configuración de la consola	Parámetro y operación API
Protección contra amenazas	Habilita o deshabilita el monitoreo de señales de actividad maliciosa o contraseñas inseguras	Cliente de aplicación o grupo de usuarios	Puede bloquear automáticamente el inicio de sesión o requerir MFA cuando los usuarios muestran indicadores de compromiso	Protección contra amenazas - Configuración de protección	SetRiskConfiguration Los parámetros de `SetRiskConfiguration` definen la configuración de protección contra amenazas.

Protección de clientes confidenciales con secretos de cliente

El secreto de cliente es una cadena opcional que se asocia a un cliente de aplicación. Todas las solicitudes de autenticación a clientes de aplicación con secretos de cliente deben incluir un hash secreto que se genera a partir del nombre de usuario, el ID de cliente y el secreto de cliente. Las personas que no conozcan el secreto del cliente quedarán excluidas de la aplicación desde el principio.

Sin embargo, los secretos de cliente tienen limitaciones. Si inserta un secreto de cliente en un software de cliente público, su secreto de cliente está abierto a la inspección. Esto abre la posibilidad de crear usuarios, enviar solicitudes de restablecimiento de contraseñas y realizar otras operaciones en el cliente de aplicación. Los secretos de cliente se deben implementar solo cuando una aplicación sea la única entidad que tiene acceso al secreto. Normalmente, esto es posible en aplicaciones cliente confidenciales en el servidor. Esto también se aplica a las aplicaciones M2M, en las que se

requiere un secreto de cliente. Guarde el secreto del cliente en un almacenamiento local cifrado o AWS Secrets Manager. Nunca permita que el secreto de cliente esté visible en la red pública de Internet.

Protección de otros secretos

Su sistema de autenticación con grupos de usuarios de Amazon Cognito puede gestionar datos privados, contraseñas y credenciales de AWS . A continuación, encontrará algunas de las prácticas recomendadas para gestionar los secretos a los que podría acceder su aplicación.

Contraseñas

Los usuarios pueden introducir contraseñas al iniciar sesión en la aplicación. Amazon Cognito tiene tokens de actualización que su aplicación puede emplear para continuar con las sesiones de usuario caducadas sin que se solicite una nueva contraseña. No coloque contraseñas ni hashes de contraseñas en el almacenamiento local. Diseñe su aplicación para que trate las contraseñas como opacas y solo las transmita a su grupo de usuarios.

Como práctica recomendada, implemente la autenticación sin contraseña con WebAuthn claves de acceso. Si debe implementar contraseñas, utilice el flujo de autenticación de contraseña remota segura (SRP) y la autenticación multifactor (MFA).

AWS credenciales

La autenticación administrativa y las operaciones administrativas del grupo de usuarios requieren la autenticación con AWS credenciales. Para implementar estas operaciones en una aplicación, conceda acceso seguro a AWS las credenciales temporales. Conceda acceso a las credenciales únicamente a las aplicaciones que se ejecutan en un componente de servidor que usted controle. No coloques aplicaciones que contengan AWS credenciales en sistemas públicos de control de versiones, por ejemplo. GitHub No codifique las AWS credenciales en aplicaciones públicas del lado del cliente.

Verificador de código PKCE

La clave de prueba para el intercambio de códigos o PKCE se usa para la concesión de códigos de autorización de OpenID Connect (OIDC) con el servidor de autorización de su grupo de usuarios. Las aplicaciones comparten los secretos de los verificadores de código con su grupo de usuarios cuando solicitan códigos de autorización. Para intercambiar códigos de autorización por tokens, los clientes deben confirmar que conocen el verificador de códigos. Esto evita la emisión de tokens con códigos de autorización interceptados.

Los clientes deben generar un nuevo verificador de códigos aleatorios con cada solicitud de autorización. El uso de un verificador de código estático o predecible significa que solo entonces se requiere que un atacante intercepte el verificador codificado y el código de autorización. Diseñe la aplicación de manera que no exponga los valores del verificador de código a los usuarios.

Privilegio mínimo de administración del grupo de usuarios

Las políticas de IAM pueden definir el nivel de acceso que tienen las entidades principales a las operaciones de administración y autenticación administrativa del grupo de usuarios de Amazon Cognito. Por ejemplo:

- A un servidor web, conceda permisos de autenticación con operaciones de API administrativas.
- A un AWS IAM Identity Center usuario que administre un grupo de usuarios en su grupo Cuenta de AWS, otorgue permisos para el mantenimiento y la generación de informes del grupo de usuarios.

El nivel de granularidad de los recursos en Amazon Cognito se limita a dos tipos de recursos a efectos de la política de IAM: grupo de usuarios y grupo de identidades. Tenga en cuenta que no puede aplicar permisos para administrar clientes de aplicación individuales. Configure los grupos de usuarios sabiendo que los permisos que conceda son efectivos en todos los clientes de aplicaciones. Si su organización tiene varios inquilinos de aplicaciones y su modelo de seguridad requiere separar las responsabilidades administrativas entre los inquilinos, implemente la tenencia múltiple con un inquilino por grupo de usuarios.

Aunque puede crear políticas de IAM con permisos para operaciones de autenticación de usuarios como `InitiateAuth`, esos permisos no surten efecto. Las operaciones de API públicas y autorizadas por token no están sujetas a los permisos de IAM. De las operaciones de autenticación del grupo de usuarios disponibles, solo puede conceder permisos a operaciones administrativas en el servidor, como `AdminInitiateAuth`.

Puede limitar los niveles de administración del grupo de usuarios con listas de `Action` de privilegios mínimos. El siguiente ejemplo de política es para un administrador que puede administrar los servidores de recursos IdPs, los clientes de aplicaciones y el dominio del grupo de usuarios, pero no los usuarios ni el grupo de usuarios.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "UserPoolClientAdministrator",
      "Action": [
        "cognito-idp:CreateIdentityProvider",
        "cognito-idp:CreateManagedLoginBranding",
        "cognito-idp:CreateResourceServer",
        "cognito-idp:CreateUserPoolDomain",
        "cognito-idp:DeleteIdentityProvider",
        "cognito-idp:DeleteResourceServer",
        "cognito-idp:DeleteUserPoolDomain",
        "cognito-idp:DescribeIdentityProvider",
        "cognito-idp:DescribeManagedLoginBranding",
        "cognito-idp:DescribeManagedLoginBrandingByClient",
        "cognito-idp:DescribeResourceServer",
        "cognito-idp:DescribeUserPool",
        "cognito-idp:DescribeUserPoolClient",
        "cognito-idp:DescribeUserPoolDomain",
        "cognito-idp:GetIdentityProviderByIdentifier",
        "cognito-idp:GetUICustomization",
        "cognito-idp:ListIdentityProviders",
        "cognito-idp:ListResourceServers",
        "cognito-idp:ListUserPoolClients",
        "cognito-idp:ListUserPools",
        "cognito-idp:SetUICustomization",
        "cognito-idp:UpdateIdentityProvider",
        "cognito-idp:UpdateManagedLoginBranding",
        "cognito-idp:UpdateResourceServer",
        "cognito-idp:UpdateUserPoolClient",
        "cognito-idp:UpdateUserPoolDomain"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE"
    }
  ]
}
```

El siguiente ejemplo de política otorga la administración y la autenticación de usuarios y grupos a una aplicación en el servidor.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "UserAdminAuthN",
      "Action": [
        "cognito-idp:AdminAddUserToGroup",
        "cognito-idp:AdminConfirmSignUp",
        "cognito-idp:AdminCreateUser",
        "cognito-idp:AdminDeleteUser",
        "cognito-idp:AdminDeleteUserAttributes",
        "cognito-idp:AdminDisableProviderForUser",
        "cognito-idp:AdminDisableUser",
        "cognito-idp:AdminEnableUser",
        "cognito-idp:AdminForgetDevice",
        "cognito-idp:AdminGetDevice",
        "cognito-idp:AdminGetUser",
        "cognito-idp:AdminInitiateAuth",
        "cognito-idp:AdminLinkProviderForUser",
        "cognito-idp:AdminListDevices",
        "cognito-idp:AdminListGroupsForUser",
        "cognito-idp:AdminListUserAuthEvents",
        "cognito-idp:AdminRemoveUserFromGroup",
        "cognito-idp:AdminResetUserPassword",
        "cognito-idp:AdminRespondToAuthChallenge",
        "cognito-idp:AdminSetUserMFAPreference",
        "cognito-idp:AdminSetUserPassword",
        "cognito-idp:AdminSetUserSettings",
        "cognito-idp:AdminUpdateAuthEventFeedback",
        "cognito-idp:AdminUpdateDeviceStatus",
        "cognito-idp:AdminUpdateUserAttributes",
        "cognito-idp:AdminUserGlobalSignOut",
        "cognito-idp:AssociateSoftwareToken",
        "cognito-idp:ListGroups",
        "cognito-idp:ListUsers",
        "cognito-idp:ListUsersInGroup",
        "cognito-idp:RevokeToken",
        "cognito-idp:UpdateGroup",
```

```
        "cognito-idp:VerifySoftwareToken"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE"
    }
  ]
}
```

Protección y verificación de tokens

Los tokens pueden contener referencias internas a la pertenencia a un grupo y a los atributos del usuario que quizá no desee revelar al usuario final. No guarde los tokens de acceso y de identificación en un almacenamiento local. Los tokens de actualización se cifran con una clave a la que solo puede acceder su grupo de usuarios, y son opacos para los usuarios y las aplicaciones. Revoque los tokens de actualización cuando los usuarios cierren sesión o cuando determine que no es deseable mantener la sesión de un usuario por motivos de seguridad.

Utilice los tokens de acceso para autorizar el acceso únicamente a los sistemas que verifiquen de forma independiente que el token es válido y no ha caducado. Para obtener recursos de verificación, consulte Verificación de un token web JSON.

Determinación de los proveedores de identidad confiables

Al configurar el grupo de usuarios con proveedores de identidad SAML u OIDC (IdPs), IdPs puede crear nuevos usuarios, establecer los atributos de los usuarios y acceder a los recursos de la aplicación. Los proveedores SAML y OIDC suelen utilizarse en situaciones entre empresas (B2B) o empresariales, en las que usted o su cliente inmediato controlan la membresía y la configuración del proveedor.

Los proveedores sociales ofrecen cuentas de usuario a cualquier usuario de Internet y están menos bajo su control que los proveedores empresariales. Activa las redes sociales IdPs en el cliente de tu aplicación solo cuando estés preparado para permitir que los clientes públicos inicien sesión y accedan a los recursos de tu aplicación.

El efecto de los ámbitos en el acceso a los perfiles de usuario

Puede solicitar ámbitos de control de acceso en sus solicitudes de autenticación al servidor de autorización del grupo de usuarios. Estos ámbitos pueden conceder a los usuarios acceso a recursos

externos, así como acceso para ver y modificar sus propios perfiles de usuario. Configure los clientes de aplicación para que admitan los ámbitos mínimos necesarios para el funcionamiento de la aplicación.

El `aws.cognito.signin.user.admin` alcance está presente en todos los tokens de acceso emitidos por la autenticación del SDK con operaciones como InitiateAuth: Está diseñado para las operaciones de autoservicio de perfiles de usuario en su aplicación. También puede solicitar este ámbito al servidor de autorización. Este ámbito es obligatorio para las operaciones autorizadas con un token, como y. UpdateUserAttributesGetUser El efecto de estas operaciones está limitado por los permisos de lectura y escritura del cliente de aplicación.

Los ámbitos `openid`, `profile`, `email` y phone autorizan las solicitudes al El punto de conexión userInfo en el servidor de autorización de su grupo de usuarios. Definen los atributos que puede devolver el punto de conexión. El ámbito `openid`, cuando se solicita sin otros ámbitos, devuelve todos los atributos disponibles; sin embargo, cuando se solicitan más ámbitos en la solicitud, la respuesta se reduce a los atributos representados por los ámbitos adicionales. El ámbito `openid` también indica una solicitud de un token de identificación; si omite este ámbito de la solicitud a su Autorizar punto de conexión, Amazon Cognito solo emitirá un token de acceso y, cuando proceda, un token de actualización. Para obtener más información, consulte los Ámbitos de OpenID Connect en Condiciones de uso de la aplicación.

Desinfección de las entradas para los atributos de usuario

Los atributos de usuario que podrían terminar como métodos de entrega y nombres de usuario, como `email`, tienen restricciones de formato. Otros atributos pueden tener tipos de datos de cadena, booleanos o numéricos. Los valores de los atributos de cadena admiten una variedad de entradas. Configure la aplicación para protegerse de los intentos de escribir datos no deseados en su directorio de usuarios o en los mensajes que Amazon Cognito envía a los usuarios. Realice una validación en el cliente de los valores de los atributos de cadena enviados por el usuario en su aplicación antes de enviarlos a Amazon Cognito.

Los grupos de usuarios asignan atributos IdPs a su grupo de usuarios en función de la asignación de atributos que especifique. Asigne únicamente los atributos de IdP seguros y predecibles a los atributos de cadena del grupo de usuarios.

Adición de MFA a un grupo de usuarios.

La MFA añade un factor de autenticación del tipo algo que poseemos al factor inicial del tipo algo que sabemos que normalmente es un nombre de usuario y una contraseña. Puede utilizar mensajes de

texto SMS, mensajes de correo electrónico o contraseñas temporales de un solo uso (TOTP) como factor adicional para el inicio de sesión de los usuarios que tengan la contraseña como el factor de autenticación principal.

Multi-factor la autenticación (MFA) aumenta la seguridad de los usuarios locales de la aplicación. En el caso de los usuarios federados, Amazon Cognito delega todos los procesos de autenticación al IdP y no les ofrece factores de autenticación adicionales.

> **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 exija un factor de autenticación adicional en respuesta a un aumento del nivel de riesgo. Para añadir la autenticación flexible a un grupo de usuarios, consulte Seguridad avanzada con protección contra amenazas.

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 de MFA. Cuando la MFA es obligatoria, debe incluir la configuración de MFA en la incorporación de usuarios para que el grupo de usuarios les permita iniciar sesión.

El inicio de sesión administrado 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, el inicio de sesión administrado no solicita nada 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.

Temas

- Cosas que debe saber acerca de la MFA de grupos de usuarios
- Preferencias de MFA del usuario
- Detalles de la lógica de MFA en tiempo de ejecución del usuario

- Configuración de un grupo de usuarios para la autenticación multifactor
- MFA con mensajes SMS y correo electrónico
- MFA con token de software TOTP

Cosas que debe saber acerca de la MFA de grupos de usuarios

Antes de configurar la MFA, tenga en cuenta lo siguiente:

- Los usuarios pueden tener MFA o iniciar sesión sin contraseña, con una excepción: las claves de paso con verificación de usuario pueden cumplir los requisitos de MFA como primer factor cuando se configuran en su grupo de usuarios. `FactorConfiguration` `MULTI_FACTOR_WITH_USER_VERIFICATION` `WebAuthnConfiguration` Las claves de paso no se pueden usar como un segundo factor para iniciar sesión con contraseña.
  - No puede configurar el MFA como obligatorio en los grupos de usuarios que admiten contraseñas de un solo uso.
  - No puede añadir `EMAIL_OTP` ni `SMS_OTP` añadir `AllowedFirstAuthFactors` cuando se requiere MFA en su grupo de usuarios. Puede añadir `WEB_AUTHN` cuándo `FactorConfiguration` está establecido en. `MULTI_FACTOR_WITH_USER_VERIFICATION`
  - Choice-basedel inicio de sesión solo ofrece `PASSWORD` y tiene `PASSWORD_SRP` en cuenta todos los clientes de aplicaciones cuando se requiere MFA en el grupo de usuarios. Para obtener más información sobre los flujos de nombre de usuario y contraseña, consulte Sign-in con contraseñas persistentes y Sign-in con contraseñas persistentes y una carga segura en el capítulo sobre autenticación de esta guía.
  - En los grupos de usuarios en los que la MFA es opcional, los usuarios que hayan configurado un factor de MFA solo pueden iniciar sesión con flujos de autenticación de nombre de usuario y contraseña en el inicio de sesión basado en opciones. Estos usuarios son aptos para todos los flujos de inicio de sesión basados en clientes.

  En la siguiente tabla, se describe el efecto de la configuración de MFA del grupo de usuarios y la configuración de los factores de MFA por parte del usuario en la capacidad de los usuarios de iniciar sesión con factores sin contraseña.

Configuración de la MFA de grupo de usuarios	Estado de MFA del usuario	Webauthn/OTP disponible	Se solicita el MFA después de iniciar sesión con contraseña	Puede iniciar sesión con WebAuthn/OTP
Obligatorio	Configured	No	Sí	No
Obligatorio	No está configurado	No	No (no puede iniciar sesión)	No
Opcional	Configured	Puede configurar WebAuthn pero no puede iniciar sesión con la clave de paso	Sí	No
Opcional	No está configurado	Sí	No	Sí
Opcional (con MFA de clave de paso habilitada)	MFA de clave de paso configurada	Sí	Sí (la clave de paso con verificación de usuario satisface la MFA de forma independiente)	Sí (la clave de paso con verificación de usuario satisface la MFA de forma independiente)
Obligatorio (con el MFA de clave de paso activado)	MFA de clave de paso configurada	Sí	Sí (la clave de paso con verificación de usuario satisface la MFA de forma independiente)	Sí (la clave de paso con verificación de usuario satisface la MFA de forma independiente)
Desactivado	Cualquiera	Sí	No	Sí

- El método de MFA preferido del usuario influye en los métodos que este pueda utilizar para recuperar la contraseña. Los usuarios cuya MFA preferida se realice por mensaje de correo

electrónico no pueden recibir un código de restablecimiento de contraseña por correo electrónico. Los usuarios cuya MFA preferida se realice por mensaje SMS no pueden recibir un código de restablecimiento de contraseña por SMS.

La configuración de la recuperación de contraseñas debe ofrecer una opción alternativa para cuando el usuario no pueda utilizar el método de restablecimiento de contraseña preferido. Por ejemplo, puede darse el caso de que sus mecanismos de recuperación tengan el correo electrónico como primera prioridad y la MFA de correo electrónico puede ser una opción en el grupo de usuarios. En este caso, añada la recuperación de SMS-message cuentas como segunda opción o utilice las operaciones administrativas de la API para restablecer las contraseñas de esos usuarios.

Amazon Cognito responde a las solicitudes de restablecimiento de contraseñas de usuarios que no disponen de un método de recuperación válido con una respuesta de error `InvalidParameterException`.

En el ejemplo del cuerpo de la solicitud se UpdateUserPoolmuestra una opción en la `AccountRecoverySetting` que los usuarios pueden recurrir a la recuperación mediante un mensaje SMS cuando el restablecimiento de la contraseña de un mensaje de correo electrónico no está disponible.

- Los usuarios no pueden recibir códigos de MFA y de restablecimiento de contraseña en la misma dirección de correo electrónico o número de teléfono. Si usan contraseñas de un solo uso (OTP) desde los mensajes de correo electrónico para la MFA, deben usar mensajes SMS para recuperar la cuenta. Si usan OTP desde los mensajes SMS para la MFA, deben usar mensajes de correo electrónico para recuperar la cuenta. En los grupos de usuarios con MFA, es posible que los usuarios no puedan completar la recuperación automática de contraseñas si tienen atributos para su dirección de correo electrónico, pero no un número de teléfono, o si su número de teléfono no tiene una dirección de correo electrónico.

  Para evitar que los usuarios no puedan restablecer sus contraseñas en los grupos de usuarios con esta configuración, defina los atributos `email` y `phone_number` según sea necesario. Si lo prefiere, puede configurar procesos que siempre recopilen y establezcan esos atributos cuando los usuarios se registren o cuando los administradores creen perfiles de usuario. Cuando los usuarios tienen ambos atributos, Amazon Cognito envía automáticamente códigos de restablecimiento de contraseñas al destino que no sea el factor de MFA del usuario.
- Si activa la MFA en el grupo de usuarios y elige Mensaje de texto SMS o Mensaje de correo electrónico como segundo factor, puede enviar mensajes a un atributo de número de teléfono o

correo electrónico que no haya verificado en Amazon Cognito. Una vez que el usuario complete la MFA, Amazon Cognito establece su atributo `phone_number_verified` o `email_verified` en `true`.

- Tras cinco intentos erróneos de presentar un código MFA, Amazon Cognito inicia el proceso de bloqueo por tiempo de espera exponencial descrito en Comportamiento de bloqueo por intentos de inicio de sesión con error.
- 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.
- Para cambiar el estado de la MFA de los usuarios en respuesta a eventos detectados con protección contra amenazas, active la MFA y configúrela como opcional en la consola del grupo de usuarios de Amazon Cognito. Para obtener más información, consulte Seguridad avanzada con protección contra amenazas.
- Los mensajes de correo electrónico y SMS requieren que los usuarios tengan los atributos de dirección de correo electrónico y número de teléfono. Puede establecer `email` o `phone_number` como atributos obligatorios en el grupo de usuarios. En ese caso, los usuarios no podrán completar el registro a menos que proporcionen un número de teléfono. Si no establece estos atributos como obligatorios, pero quiere ejecutar una MFA por correo electrónico o mensaje SMS, pida a los usuarios su dirección de correo electrónico o número de teléfono cuando se registren. Le recomendamos que configure el grupo de usuarios para que envíe mensajes automáticamente a los usuarios para comprobar estos atributos.

  Amazon Cognito considera verificados un número de teléfono o una dirección de correo electrónico si un usuario ha recibido correctamente un código temporal por SMS o mensaje de correo electrónico y lo ha devuelto en una solicitud de VerifyUserAttributeAPI. Como alternativa, su equipo puede configurar números de teléfono y marcarlos como verificados con una aplicación administrativa que realice solicitudes de AdminUpdateUserAttributesAPI.
- Si ha configurado que se exija una MFA y ha activado más de un factor de autenticación, Amazon Cognito pedirá a los nuevos usuarios que seleccionen el factor de MFA que deseen usar. Los usuarios deben tener un número de teléfono para configurar la MFA de mensajes SMS y una dirección de correo electrónico para configurar la MFA de mensajes de correo electrónico. Si un usuario no tiene definido el atributo para ninguna MFA basada en mensajes disponible, Amazon Cognito le pedirá que configure la MFA con TOTP. La pregunta para elegir un factor de

MFA (`SELECT_MFA_TYPE`) y configurar un factor elegido (`MFA_SETUP`) se presenta como una respuesta a un desafío InitiateAuthy a las operaciones de la AdminInitiateAuthAPI.

Preferencias de MFA del usuario

Los usuarios pueden configurar varios factores de MFA. Solo uno puede estar activo. Puede elegir la preferencia de MFA efectiva para los usuarios en la configuración del grupo de usuarios o en las peticiones de los usuarios. Un grupo de usuarios pide a un usuario los códigos de MFA cuando la configuración del grupo de usuarios y su propia configuración de usuario cumplen las siguientes condiciones:

1. La MFA está establecida como opcional u obligatoria en el grupo de usuarios.
2. El usuario tiene un atributo `email` o `phone_number` válido o ha configurado una aplicación de autenticación para la TOTP.
3. Al menos un factor de la MFA está activo.
4. Se ha establecido un factor de MFA como preferido.

Evite el uso del mismo factor para el inicio de sesión y la MFA

Es posible configurar el grupo de usuarios de forma que un solo factor de inicio de sesión sea la única opción de inicio de sesión y MFA disponible para algunos o todos los usuarios. Este resultado puede producirse cuando el principal caso de uso del inicio de sesión son los mensajes de correo electrónico o SMS-message las contraseñas de un solo uso (OTP). El MFA preferido de un usuario puede ser el mismo tipo de factor que su inicio de sesión en las siguientes condiciones:

- Se requiere MFA en el grupo de usuarios.
- El correo electrónico y los SMS OTP están disponibles como opciones de inicio de sesión y MFA en el grupo de usuarios.
- El usuario inicia sesión con el OTP de correo electrónico o mensaje SMS.
- Tienen un atributo de dirección de correo electrónico pero no un atributo de número de teléfono, o un atributo de número de teléfono pero no un atributo de dirección de correo electrónico.

En este escenario, el usuario puede iniciar sesión con una OTP de correo electrónico y completar la MFA con una OTP de correo electrónico. Esta opción anula la función esencial de la MFA. Los usuarios que inicien sesión con contraseñas de un solo uso deben poder utilizar métodos de entrega

diferentes a los de la MFA. Cuando los usuarios tienen opciones de SMS y correo electrónico, Amazon Cognito asigna automáticamente un factor diferente. Por ejemplo, cuando un usuario inicia sesión con una OTP de correo electrónico, su MFA preferida es la OTP de SMS.

Siga los siguientes pasos para abordar la autenticación del mismo factor cuando su grupo de usuarios admita la autenticación OTP tanto para el inicio de sesión como para la MFA.

1. Habilite la OTP por correo electrónico y SMS como factores de inicio de sesión.
2. Habilite tanto el correo electrónico como el SMS OTP como factores de MFA.
3. Recopilación

Configuración del grupo de usuarios y su efecto en las opciones de MFA

La configuración del grupo de usuarios influye en los métodos de MFA que los usuarios pueden elegir. A continuación, se muestran algunos ajustes del grupo de usuarios que influyen en la capacidad de los usuarios para configurar la MFA.

- En la configuración de Multi-factor autenticación del Sign-inmenú de la consola de Amazon Cognito, puede configurar el MFA como opcional o obligatorio, o bien desactivarlo. El equivalente en API de este ajuste es el MfaConfigurationparámetro de `CreateUserPool`, `UpdateUserPool`, y. `SetUserPoolMfaConfig`

  Además, en la configuración de Multi-factor autenticación, la configuración de los métodos de MFA determina los factores de MFA que los usuarios pueden configurar. El equivalente en API de esta configuración es la SetUserPoolMfaConfigoperación.
- En el Sign-inmenú, en Recuperación de cuentas de usuario, puede configurar la forma en que su grupo de usuarios envía mensajes a los usuarios que olvidan su contraseña. El método de MFA de un usuario no puede tener el mismo método de entrega de MFA que el método de entrega de grupos de usuarios para los códigos de contraseña olvidada. El parámetro de la API para el método de entrega de contraseñas olvidadas es el AccountRecoverySettingparámetro de y. `CreateUserPool` `UpdateUserPool`

  Por ejemplo, los usuarios no pueden configurar la MFA de correo electrónico si la opción de recuperación es Solo correo electrónico. Esto se debe a que no puede habilitar la MFA del correo electrónico y configurar la opción de recuperación en Solo correo electrónico en el mismo grupo de usuarios. Si establece esta opción en Correo electrónico si está disponible, de lo contrario, SMS, el correo electrónico es la opción de recuperación prioritaria, pero su grupo de usuarios puede recurrir a los mensajes SMS cuando un usuario no reúna los requisitos para la recuperación

por mensaje de correo electrónico. En este escenario, los usuarios pueden configurar la MFA por correo electrónico como preferida y solo pueden recibir un mensaje SMS cuando intenten restablecer su contraseña.

- Si establece solo un método de MFA como disponible, no necesita administrar las preferencias de MFA del usuario.
- Una configuración de SMS activa convierte automáticamente los mensajes SMS en un método de MFA disponible en su grupo de usuarios.

  Una configuración de correo electrónico activa con sus propios recursos de Amazon SES en un grupo de usuarios y el plan de características Essentials o Plus convierte automáticamente los mensajes de correo electrónico en un método de MFA disponible en el grupo de usuarios.
- Al configurar la MFA como obligatoria en un grupo de usuarios, los usuarios no pueden habilitar ni deshabilitar ningún método de MFA. Solo puede establecer un método preferido.
- Al configurar la MFA como opcional en un grupo de usuarios, el inicio de sesión administrado no pide a los usuarios que configuren la MFA, pero sí les pide un código de MFA cuando tienen un método de MFA preferido.
- Al activar la protección contra amenazas y configurar las respuestas de autenticación flexible en el modo de función completa, la MFA debe ser opcional en su grupo de usuarios. Una de las opciones de respuesta con la autenticación flexible consiste en exigir la MFA a un usuario cuyo intento de inicio de sesión se considere que contiene un nivel de riesgo.

  La configuración de atributos obligatorios del Sign-upmenú de la consola determina si los usuarios deben proporcionar una dirección de correo electrónico o un número de teléfono para registrarse en la aplicación. Los mensajes de correo electrónico y SMS se convierten en factores de MFA aptos cuando un usuario tiene el atributo correspondiente. El parámetro Schema de `CreateUserPool` establece los atributos obligatorios.
- Cuando configura la MFA como obligatoria en un grupo de usuarios y un usuario inicia sesión con el inicio de sesión administrado, Amazon Cognito le pide que seleccione un método de MFA entre los métodos disponibles para su grupo de usuarios. El inicio de sesión administrado gestiona la recopilación de una dirección de correo electrónico o un número de teléfono y la configuración de la TOTP. El siguiente diagrama muestra la lógica de las opciones que Amazon Cognito presenta a los usuarios.

Configuración de las preferencias de MFA para los usuarios

Puede configurar las preferencias de la MFA para los usuarios en un modelo de autoservicio con autorización de token de acceso o en un modelo administrado por el administrador con operaciones de API administrativas. Estas operaciones habilitan o deshabilitan los métodos de la MFA y establecen uno de los diversos métodos como opción preferida. Una vez que el usuario haya establecido una preferencia de MFA, cuando inicie sesión, Amazon Cognito le pedirá que proporcione un código del método de MFA que prefiera. A los usuarios que no hayan establecido una preferencia se les pedirá que elijan un método preferido en un desafío `SELECT_MFA_TYPE`.

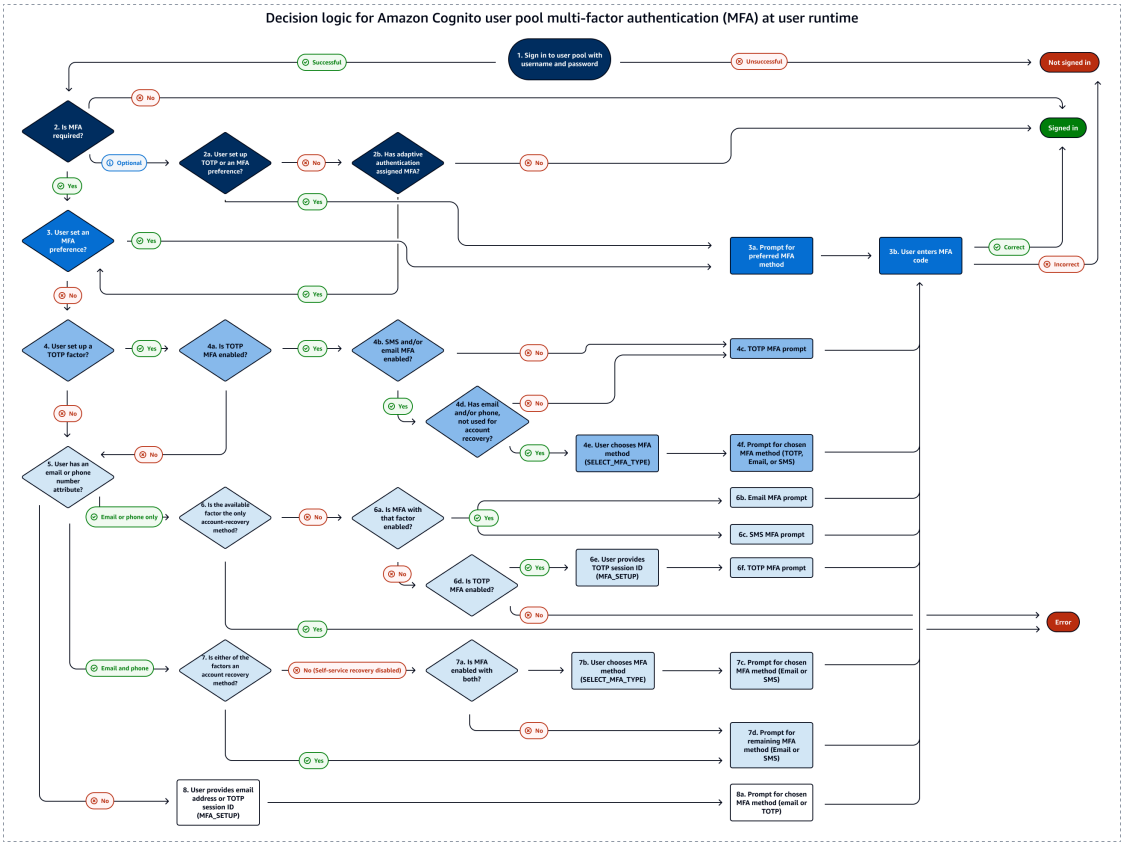
- En un modelo de autoservicio de usuario o en una aplicación pública SetUserMfaPreference, autorizada con el token de acceso de un usuario que ha iniciado sesión, establece la configuración de MFA.
- En una aplicación confidencial o gestionada por el administrador, autorizada con AWS credenciales administrativas AdminSetUserPreference, establece la configuración de MFA.

También puede configurar las preferencias de la MFA del usuario desde el menú Usuarios de la consola de Amazon Cognito. Para obtener más información sobre los modelos de autenticación pública y confidencial en la API de grupos de usuarios de Amazon Cognito, consulte Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

Detalles de la lógica de MFA en tiempo de ejecución del usuario

A fin de determinar los pasos que deben seguirse cuando los usuarios inician sesión, su grupo de usuarios evalúa las preferencias de MFA de los usuarios, los atributos de los usuarios, la configuración de MFA del grupo de usuarios, las acciones de protección contra amenazas y la configuración de la recuperación automática de cuentas. A continuación, inicia sesión en los usuarios, les pide que elijan un método de MFA, les pide que configuren un método de MFA o les pide que utilicen MFA. Para configurar un método de MFA, los usuarios deben proporcionar una dirección de correo electrónico o un número de teléfono o registrar un autenticador TOTP. También pueden configurar las opciones de MFA y registrar una opción preferida por adelantado. El siguiente diagrama muestra los efectos detallados de la configuración del grupo de usuarios en los intentos de inicio de sesión inmediatamente después del registro inicial.

La lógica que se muestra aquí se aplica a SDK-based las aplicaciones y a los inicios de sesión gestionados, pero es menos visible en los inicios de sesión gestionados. Al solucionar problemas de MFA, retroceda desde los resultados de los usuarios hasta las configuraciones del perfil de usuario y del grupo de usuarios que contribuyeron a la decisión.



La siguiente lista corresponde a la numeración del diagrama de lógica de decisiones y describe cada paso en detalle. Un



indica una autenticación correcta y la conclusión del flujo. Un



indica que la autenticación no se ha realizado correctamente.

1. Un usuario presenta su nombre de usuario o nombre de usuario y contraseña en la pantalla de inicio de sesión. Si no presenta credenciales válidas, se deniega su solicitud de inicio de sesión.
2. Si se realiza correctamente la autenticación con nombre de usuario y contraseña, determine si el MFA es obligatorio, opcional o si está desactivado. Si está desactivado, el nombre de usuario y la contraseña correctos permiten que la autenticación se realice correctamente.

   a. Si la MFA es opcional, determine si el usuario ha configurado previamente un autenticador TOTP. Si ha configurado un TOTP, solicite la MFA con

TOTP. Si responde correctamente al desafío de la MFA, iniciará sesión.



b. Determine si la característica de autenticación flexible de la protección contra amenazas ha requerido que el usuario configure la MFA. Si no ha asignado una MFA, el usuario ha iniciado sesión.



3. Si se requiere la MFA o si la autenticación flexible tiene una MFA asignada, determine si el usuario ha establecido un factor de MFA como habilitado y preferido. Si lo han hecho, solicite una MFA con ese factor. Si responde correctamente al desafío de la MFA, iniciará sesión.



4. Si el usuario no ha establecido una preferencia de MFA, determine si ha registrado un autenticador TOTP.
   a. Si el usuario ha registrado un autenticador TOTP, determine si la MFA con TOTP está disponible en el grupo de usuarios (se puede deshabilitar después de que los usuarios hayan configurado previamente los autenticadores).
   b. Determine si el mensaje de correo electrónico o el SMS-message MFA también están disponibles en el grupo de usuarios.
   c. Si no está disponible la MFA por correo electrónico ni por SMS, solicite al usuario la MFA con TOTP. Si responde correctamente al desafío de la MFA, iniciará sesión.



   d. Si el correo electrónico o la MFA por SMS están disponibles, determine si el usuario tiene el atributo `email` o `phone_number` correspondiente. Si es así, tendrá a su disposición cualquier atributo que no sea el método principal de recuperación de cuentas de autoservicio y que esté habilitado para la MFA.
   e. Plantee un desafío `SELECT_MFA_TYPE` al usuario con opciones `MFAS_CAN_SELECT` que incluyan el TOTP y los factores de MFA disponibles por SMS o correo electrónico.
   f. Pídale al usuario el factor que ha seleccionado en respuesta al desafío `SELECT_MFA_TYPE`. Si responde correctamente al desafío de la MFA, iniciará sesión.



5. Si el usuario no ha registrado un autenticador TOTP o si lo ha hecho, pero el TOTP con MFA está deshabilitado actualmente, determine si el usuario tiene un atributo `email` o `phone_number`.

6. Si el usuario solo tiene una dirección de correo electrónico o solo un número de teléfono, determine si ese atributo también es el método que implementa el grupo de usuarios para enviar mensajes de recuperación de cuentas para restablecer la contraseña. Si es verdadero, no puede completar el inicio de sesión con el MFA obligatorio y Amazon Cognito devuelve un error. Para activar el inicio de sesión para este usuario, debe añadir un atributo que no sea de recuperación o registrar un autenticador TOTP para el usuario.



   a. Si tiene una dirección de correo electrónico o un número de teléfono disponibles que no sean de recuperación, determine si el factor MFA de correo electrónico o SMS correspondiente está activado.

   b. Si tiene un atributo de dirección de correo electrónico no recuperable y la MFA de correo electrónico está habilitada, envíeles un desafío `EMAIL_OTP`. Si responde correctamente al desafío de la MFA, iniciará sesión.



   c. Si tiene un atributo de número de teléfono no recuperable y la MFA de SMS está habilitada, envíeles un desafío `SMS_MFA`. Si responde correctamente al desafío de la MFA, iniciará sesión.



   d. Si no tienen un atributo que sea apto para un factor de MFA por correo electrónico o SMS habilitado, determine si la MFA con TOTP está habilitada. Si la MFA con TOTP está habilitada, no puede completar el inicio de sesión con la MFA obligatoria y Amazon Cognito devuelve un error. Para activar el inicio de sesión para este usuario, debe añadir un atributo que no sea de recuperación o registrar un autenticador TOTP para el usuario.



      > **Note**
      >
      > Este paso ya se evaluó como No si el usuario tiene un autenticador TOTP pero la MFA con TOTP está deshabilitada.

   e. Si la MFA con TOTP está activada, presente al usuario un desafío `MFA_SETUP` con `SOFTWARE_TOKEN_MFA` en las opciones `MFAS_CAN_SETUP`. Para completar este desafío, debe registrar por separado un autenticador TOTP para el usuario y responder con `"ChallengeName": "MFA_SETUP", "ChallengeResponses": {"USERNAME": "[username]", "SESSION": "[Session ID from VerifySoftwareToken]}"`.

f. Cuando el usuario responda al `MFA_SETUP` desafío con el token de sesión de una VerifySoftwareTokensolicitud, solicítele un desafío. `SOFTWARE_TOKEN_MFA` Si responde correctamente al desafío de la MFA, iniciará sesión.



7. Si el usuario tiene una dirección de correo electrónico y un número de teléfono, determine qué atributo, si lo hay, es el método principal de los mensajes de recuperación de la cuenta para restablecer la contraseña.

   a. Si la recuperación de cuentas de autoservicio está deshabilitada, se puede usar cualquiera de los atributos para la MFA. Determine si uno o ambos factores de MFA de correo electrónico y SMS están habilitados.

   b. Si ambos atributos están habilitados como factor de MFA, plantee al usuario un desafío `SELECT_MFA_TYPE` con las opciones `SMS_MFA` y `EMAIL_OTP` de `MFAS_CAN_SELECT`.

   c. Pídale al usuario el factor que ha seleccionado en respuesta al desafío `SELECT_MFA_TYPE`. Si responde correctamente al desafío de la MFA, iniciará sesión.



   d. Si solo un atributo es un factor de MFA elegible, preséntele un desafío para el factor restante. Si responde correctamente al desafío de la MFA, iniciará sesión.



      Esto ocurre en las siguientes situaciones:

      i. Cuando el usuario tiene atributos `email` y `phone_number`, la MFA por SMS y correo electrónico está habilitada y el método principal de recuperación de la cuenta es por correo electrónico o mensaje SMS.

      ii. Cuando tiene atributos `email` y `phone_number`, solo se habilita la MFA por SMS o correo electrónico y la recuperación automática de cuentas está deshabilitada.

8. Si el usuario no ha registrado un autenticador TOTP y no tiene un atributo `email` ni `phone_number`, póngales el desafío `MFA_SETUP`. La lista en `MFAS_CAN_SETUP` incluye todos los factores de MFA habilitados en el grupo de usuarios que no son la opción principal de recuperación de cuentas. Puede responder a este desafío con `ChallengeResponses` para la MFA con correo electrónico o TOTP. Para configurar la MFA por SMS, añada un atributo de número de teléfono por separado y reinicie la autenticación.

Para la MFA con TOTP, responda con `"ChallengeName": "MFA_SETUP", "ChallengeResponses": {"USERNAME": "[username]", "SESSION": "[Session ID from VerifySoftwareToken]"}`.

Para la MFA por correo electrónico, responda con `"ChallengeName": "MFA_SETUP", "ChallengeResponses": {"USERNAME": "[username]", "email": "[user's email address]"}`.

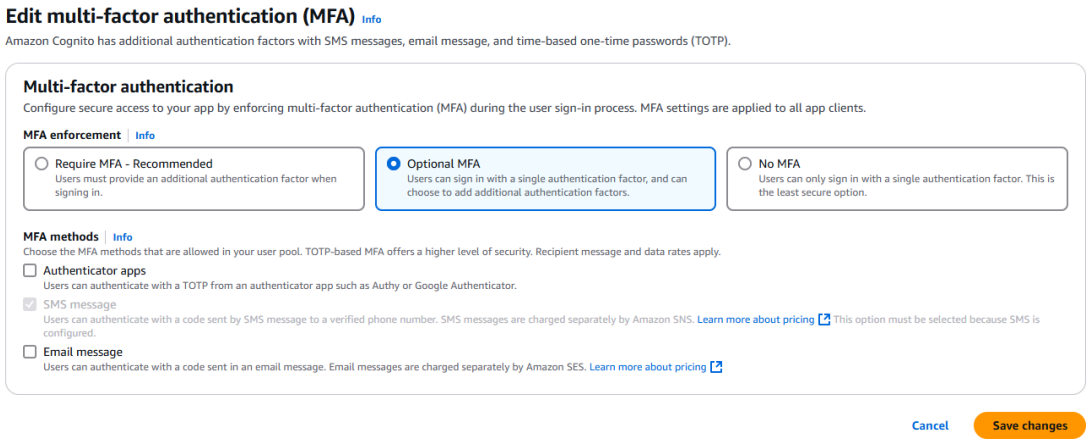
a. Pídale al usuario el factor que ha seleccionado en respuesta al desafío `SELECT_MFA_TYPE`. Si responde correctamente al desafío de la MFA, iniciará sesión.

Configuración de un grupo de usuarios para la autenticación multifactor

Puede configurar la MFA en la consola de Amazon Cognito o con la operación de SetUserPoolMfaConfigla API y los métodos del SDK.

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. Seleccione el menú Sign-in. Localice la Multi-factorautenticación y seleccione 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 código adicional de SMS, correo electrónico o contraseña temporal de un solo uso (TOTP) como factor de autenticación adicional.

b. 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 Seguridad avanzada con protección contra amenazas.

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 configurar las aplicaciones de mensajes de correo electrónico, mensajes SMS o TOTP-generating autenticadoras como segundo factor.

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 el menú Métodos de autenticación de su grupo de usuarios, busque SMS y seleccione 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.

   Si utiliza mensajes de correo electrónico como segundo factor y no ha configurado una identidad de origen para utilizar con Amazon Simple Email Service (Amazon SES) para mensajes de correo electrónico, cree una en la consola. Debe elegir la opción Enviar correo electrónico con SES. En el menú Métodos de autenticación de su grupo de usuarios, busque Correo electrónico y seleccione Editar. Seleccione una Dirección de correo electrónico del REMITENTE entre las identidades verificadas disponibles en la lista. Si elige un dominio verificado, como `example.com`, también debe configurar un Nombre del REMITENTE en el dominio verificado, como `admin-noreply@example.com`.

8. Seleccione Save changes (Guardar cambios).

MFA con mensajes SMS y correo electrónico

Los mensajes de la MFA de correo electrónico y SMS confirman que los usuarios tienen acceso a un destino de mensajes antes de poder iniciar sesión. Confirman que no solo tienen acceso a una contraseña, sino también a los mensajes SMS o a la bandeja de entrada del usuario original. Amazon Cognito solicita al usuario que proporcione el código corto que el grupo de usuarios le ha enviado después de que haya introducido correctamente un nombre de usuario y una contraseña.

La MFA con SMS y correo electrónico no requiere configuración adicional después de que el usuario añada una dirección de correo electrónico o un número de teléfono a su perfil. Amazon Cognito puede enviar mensajes a direcciones de correo electrónico y números de teléfono no verificados. Cuando un usuario completa su primera MFA, Amazon Cognito marca su dirección de correo electrónico o número de teléfono como verificados.

La autenticación del tipo de MFA comienza cuando un usuario con MFA introduce el nombre de usuario y contraseña en la aplicación. La aplicación envía estos parámetros iniciales en un método del SDK que invoca una solicitud InitiateAutho AdminInitiateAuthuna solicitud de API. `ChallengeParameters`, en la respuesta de la API, incluye un valor `CODE_DELIVERY_DESTINATION` que indica dónde se ha enviado el código de autorización. En su aplicación, muestre un formulario que pida al usuario que consulte el teléfono e incluya un elemento de entrada para introducir el código. Cuando este escriba el código, envíelo en una solicitud de la API de desafío-respuesta para completar el proceso de inicio de sesión.

Cuando un usuario con MFA inicia sesión con el nombre de usuario y la contraseña en las páginas del inicio de sesión administrado, se les solicita automáticamente el código de MFA.

Los grupos de usuarios envían mensajes SMS para notificaciones de la MFA y otras notificaciones de Amazon Cognito con los recursos de Amazon Simple Notification Service (Amazon SNS) a su Cuenta de AWS. Del mismo modo, los grupos de usuarios envían mensajes de correo electrónico con los recursos de Amazon Simple Email Service (Amazon SES) de su cuenta. Estos servicios enlazados incurren en sus propios costes en la AWS factura de envío de mensajes. También tienen requisitos adicionales para el envío de mensajes en volúmenes de producción. Para obtener más información, consulte los siguientes enlaces:

- Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito
- Precios de los mensajes SMS a nivel mundial
- Configuración de correo electrónico para grupos de usuarios de Amazon Cognito
- Precios de Amazon SES

Consideraciones sobre la MFA de SMS y mensajes de correo electrónico

- Para permitir que los usuarios inicien sesión mediante una MFA de correo electrónico, el grupo de usuarios debe tener las siguientes opciones de configuración:
  1. Tiene el plan de características Plus o Essentials en su grupo de usuarios. Para obtener más información, consulte Planes de características de grupo de usuarios.

2. El grupo de usuarios envía mensajes de correo electrónico con los propios recursos de Amazon SES. Para obtener más información, consulte Configuración de email de Amazon SES.

- El código de la MFA es válido para la opción Duración de la sesión del flujo de autenticación que ha configurado para el cliente de aplicación.

  Configure la duración de una sesión de flujo de autenticación en la consola de Amazon Cognito, en el menú Clientes de aplicación, cuando edite su cliente de aplicación. 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 Un ejemplo de sesión de autenticación.
- Cuando un usuario proporciona correctamente un código de un mensaje SMS o de correo electrónico que Amazon Cognito ha enviado a un número de teléfono o dirección de correo electrónico no verificados, Amazon Cognito marca el atributo correspondiente como verificado.
- Para que un usuario introduzca un cambio de autoservicio en el valor de un número de teléfono o una dirección de correo electrónico asociados a la MFA, debe iniciar sesión y autorizar la solicitud con un token de acceso. Si no puede acceder a su número de teléfono o dirección de correo electrónico actuales, no podrá iniciar sesión. Tu equipo debe cambiar estos valores con AWS las credenciales de administrador en las solicitudes de la AdminUpdateUserAttributesAPI.
- Después de configurar los SMS en el grupo de usuarios, no podrá deshabilitar los mensajes SMS como factor de MFA disponible.

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 el inicio de sesión administrado 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. En el 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 el inicio de sesión administrado, 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.

> **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 al iniciar sesión gestionado y registrarse en el TOTP. Para crear una regla que tenga una acción CAPTCHA y no afecte al TOTP del inicio de sesión administrado, consulte Configurando su AWS WAF ACL web para inicio de sesión gestionado TOTP MFA. Para obtener más información sobre ACL web de AWS WAF y Amazon Cognito, consulte Asocie un AWS WAF ACL web a un grupo de usuarios.

Para implementar el MFA de TOTP en una interfaz de usuario personalizada con AWS un SDK y la API de grupos de usuarios de Amazon Cognito, consulte. Configuración de MFA con TOTP para un usuario

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 del inicio de sesión administrado 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 el inicio de sesión administrado 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 el inicio de sesión administrado hasta que haya configurado la MFA. En lugar del token de acceso, puede usar el valor de `session` respuesta de una `MFA_SETUP` impugnación 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 hash. HMAC-SHA1 Los códigos generados con el SHA-256 hash devuelven un error. `Code mismatch`

Configuración de MFA con TOTP para un usuario

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 el inicio de sesión administrado presentan el TOTP a Amazon Cognito en las respuestas al desafío de MFA.

En algunas circunstancias, el inicio de sesión administrado solicita a los nuevos usuarios que configuren un autenticador TOTP. Para obtener más información, consulte Detalles de la lógica de MFA en tiempo de ejecución del usuario.

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 introducir la clave en una TOTP-generating aplicación como Google Authenticator, ya sea escaneando el código QR que la aplicación genera a partir de la clave privada o introduciéndola manualmente.
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 TOTP-generating aplicación 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. Para eliminar el token de software TOTP de su usuario, utilice. AdminDeleteSoftwareToken Tras eliminar el token de software del usuario, este ya no podrá iniciar sesión con TOTP MFA y podrá registrar un nuevo token de software. Usa esta operación cuando el usuario pierda el acceso a su TOTP-

generating dispositivo, por ejemplo, si pierde o se restablece el teléfono. Cuando su grupo de usuarios requiere MFA y no hay ningún otro factor disponible, Amazon Cognito devuelve `MFA_SETUP` un desafío la próxima vez que el usuario inicie sesión. Cuando el usuario tiene otro factor disponible, como el MFA de un mensaje SMS, el inicio de sesión recurre a ese factor.

Si el usuario sigue teniendo acceso a su TOTP-generating dispositivo, no es necesario que elimines su token de software. Amazon Cognito disocia el token de software existente cuando el usuario verifica uno nuevo. Para desactivar el MFA TOTP para su usuario, llame para modificar su usuario para SetUserMFAPreferenceque 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. SetUserMFAPreference
3. En la aplicación, comunique al usuario que ha desactivado la MFA y pídale que vuelva a iniciar sesión.

Configurando su AWS WAF ACL web para inicio de sesión gestionado TOTP MFA

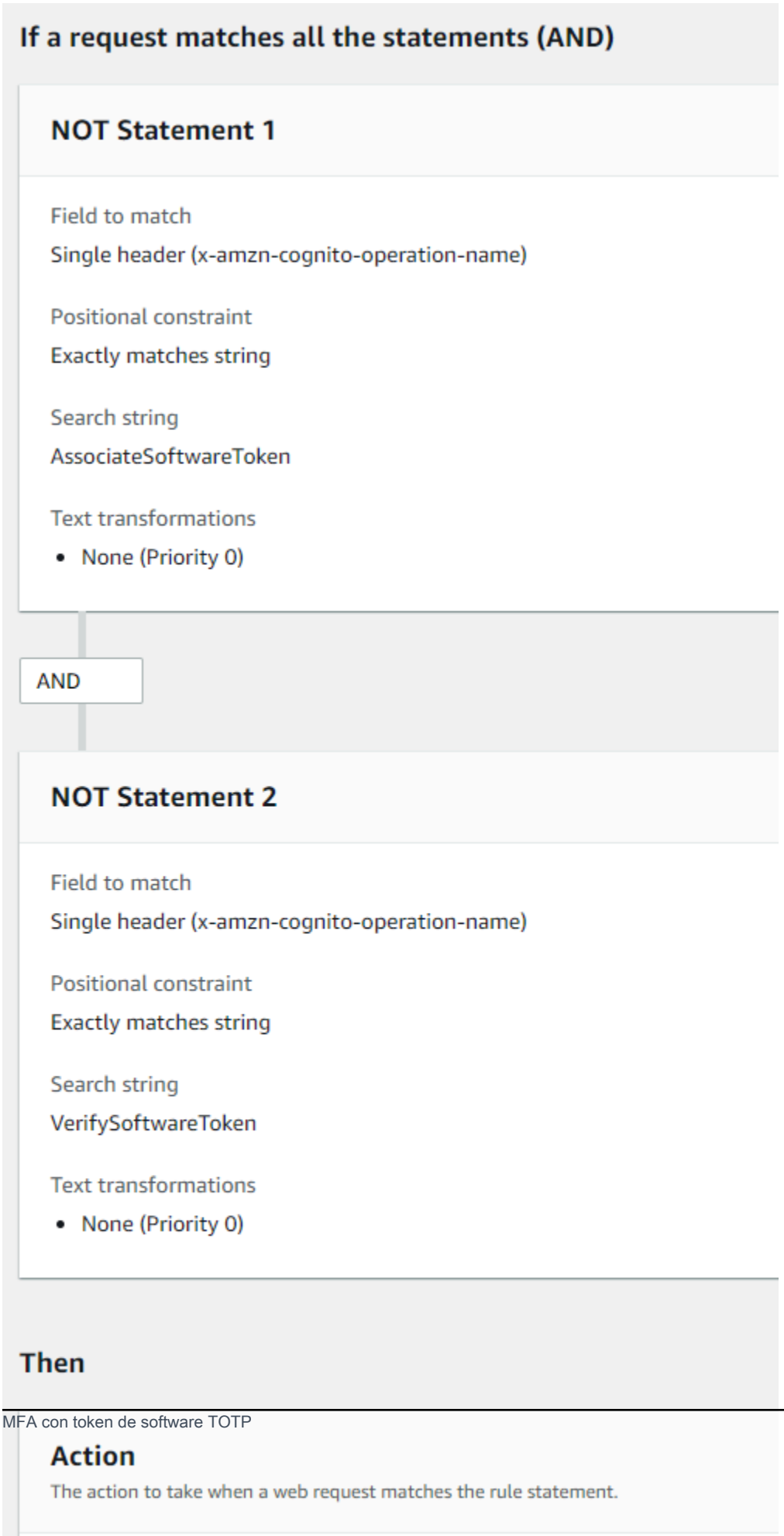
Cuando tiene una ACL AWS WAF web asociada a un grupo de usuarios y una regla de su ACL web presenta un CAPTCHA, esto puede provocar un error irrecuperable al iniciar sesión gestionado y registrarse en el TOTP. AWS WAF Las reglas de CAPTCHA solo tienen este efecto en el MFA TOTP en el inicio de sesión gestionado y en la interfaz de usuario alojada clásica. 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 en el inicio de sesión administrado, excluya los valores del encabezado `x-amzn-cognito-operation-name` de `AssociateSoftwareToken` y `VerifySoftwareToken` de la acción CAPTCHA en su 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
Action
The action to take when a web request matches the rule statement.

Para obtener más información sobre las ACL AWS WAF web y Amazon Cognito, consulte. Asocie un AWS WAF ACL web a un grupo de usuarios

Niveles de autenticación con afirmaciones de ACR y AMR

Amazon Cognito puede registrar el nivel de autenticación de un usuario y los métodos que ha utilizado. Usted proporciona un nivel de autenticación objetivo como entrada cuando un usuario inicia sesión, y Amazon Cognito informa del nivel alcanzado por el usuario y los métodos que ha completado como reclamaciones en los tokens que emite. Estas afirmaciones se aplican a cualquier inicio de sesión, incluido el inicio de sesión inicial de un usuario.

Solicita un nivel de autenticación con el `acr_values` parámetro de los puntos finales de inicio de sesión, autorización e inicio de sesión administrados, o con el `TARGET_ACR_VALUES` parámetro equivalente en el flujo de la `USER_AUTH` API. Ambos aceptan una lista ordenada de niveles de autenticación objetivo. Puede añadir el `max_age` parámetro opcional (`MAX_AGE`en la API) para exigir que el usuario se autentique dentro de un período de tiempo reciente. En respuesta a estas entradas, Amazon Cognito añade dos reclamos a los tokens que emite:

- La `acr` afirmación es una cadena que indica el nivel de autenticación que satisfizo el usuario, por ejemplo`urn:cognito:loa:3`.
- La `amr` afirmación es un conjunto de cadenas que indica los métodos de autenticación que ha completado el usuario, por ejemplo`["pwd", "otp", "mfa"]`.

Las `amr` afirmaciones `acr` y siguen las definiciones de la especificación OpenID Connect Core. Amazon Cognito utiliza sus propios URI de nivel ACR. Puede traducir los niveles de Amazon Cognito a cualquier estándar que requiera su aplicación.

Step-up la autenticación es un caso de uso común para estas afirmaciones. En la autenticación progresiva, la aplicación requiere que el usuario se autentique con un método más seguro antes de realizar una operación confidencial. Para decidir si se debe realizar una solicitud, la aplicación compara el `acr` nivel y la `auth_time` afirmación del token de identificación del usuario con el nivel y la actualidad que requiere la operación, y solicita el aumento solo cuando aún no se haya cumplido. Cuando sea necesario dar un paso adelante, redirige al usuario al punto de inicio de sesión, autorización o inicio de sesión gestionado`acr_values`, o llama al flujo de `USER_AUTH` API con. `TARGET_ACR_VALUES` Para ver un ejemplo, consulta el escenario de Step-up autenticación.

Puede solicitar ACR y AMR desde cualquier punto de entrada de inicio de sesión de Amazon Cognito. La mecánica de cada punto de entrada se documenta con ese punto de entrada.

- Inicio de sesión en el SDK y la API: solicite un nivel objetivo en el `USER_AUTH` flujo con los `MAX_AGE` parámetros `TARGET_ACR_VALUES` y. Para ver los parámetros y el comportamiento de aceleración, consulte. Step-up autenticación con el flujo USER_AUTH
- Inicio de sesión gestionado e interfaz de usuario alojada: transfiera los `max_age` parámetros `acr_values` y a los puntos finales de autorización e inicio de sesión. Para obtener más información, consulte el punto final de autorización y el punto final de inicio de sesión.
- Inicio de sesión federado: Amazon Cognito obtiene los valores de ACR y AMR de un usuario federado a partir de la respuesta del proveedor de identidad (IdP) y los asigna al campo. `AcrMapping` Para obtener información sobre Step-up autenticación para usuarios federados los proveedores de identidad del OIDC, consulte.

Para solicitar una autenticación intensificada se requiere el plan de funciones Essentials o Plus. Para obtener más información, consulte Requisitos del plan de funciones.

Temas

- Niveles y métodos de autenticación
- Configuración de los nombres de los niveles de ACR
- Requisitos del plan de funciones
- Sus responsabilidades a la hora de aumentar la autenticación

Niveles y métodos de autenticación

Amazon Cognito evalúa los factores de autenticación que completa un usuario y expresa el resultado como un nivel de autenticación en la `acr` declaración y una lista de métodos de autenticación en la notificación. `amr`

Niveles de autenticación ACR

Amazon Cognito define cuatro niveles de autenticación fijos. Las combinaciones de factores de autenticación que satisfacen cada nivel son fijas y no se pueden cambiar. Solo puede personalizar el nombre URI de cada nivel para su grupo de usuarios. Para obtener más información, consulte Configuración de los nombres de los niveles de ACR.

Amazon Cognito asigna el nivel de autenticación más alto que satisfagan plenamente los factores completados por el usuario.

Niveles de ACR y combinaciones de factores satisfactorias

Nivel ACR	Valor ACR predeterminado	Combinaciones de factores satisfactorias
1	`urn:cognito:loa:1`	Solo contraseña
2	`urn:cognito:loa:2`	Una única contraseña de un solo uso entregada por SMS o correo electrónico. El nivel 2 no incluye las claves de acceso
3	`urn:cognito:loa:3`	Una clave de paso (de plataforma o de itinerancia) propia; o una contraseña con un código de un solo uso por SMS; o una contraseña con un código de un solo uso por correo electrónico
4	`urn:cognito:loa:4`	Una contraseña con una contraseña de un solo uso (TOTP) basada en el tiempo de una aplicación de autenticación

El nivel 1 representa la autenticación solo con contraseña. La federación no satisface ninguno de los niveles de esta tabla. En cambio, Amazon Cognito deriva el nivel de autenticación de un usuario federado a partir de la respuesta del proveedor de identidad (IdP). Para obtener más información, consulte Step-up autenticación para usuarios federados.

Valores de AMR

La `amr` afirmación es una lista de cadenas. Cada factor de autenticación que completa el usuario añade su valor a la lista. Amazon Cognito puede añadir los siguientes valores.

Valores de AMR

Método de autenticación	Valor AMR	Notas
Contraseña	`pwd`	Password-family inicio de sesión, incluida la contraseña remota segura (SRP)

Método de autenticación	Valor AMR	Notas
TOTP desde una aplicación de autenticación	`otp`	Una contraseña de un solo uso basada en el tiempo de una aplicación, como una aplicación de autenticación
Código de un solo uso por SMS	`sms`	Un código de un solo uso enviado por SMS
Código de un solo uso por correo	`email_otp`	Un código de un solo uso enviado por correo electrónico
Clave de paso con un autenticador de plataforma	`swk`	Una clave vinculada a un software, por ejemplo, una clave de paso asegurada por un dispositivo biométrico
Clave de paso con un autenticador móvil	`hwk`	Una clave vinculada al hardware, por ejemplo, una clave de seguridad
Multi-factor combinación	`mfa`	Se agrega cuando el usuario completa dos o más factores distintos
Inicio de sesión federado	`fed`	Un marcador de posición predeterminado que Amazon Cognito añade cuando un usuario se federa a través de un IdP que no admite ACR ni AMR, o no devuelve valores que Amazon Cognito reconozca

Declaraciones sobre los tokens ACR y AMR

Amazon Cognito añade los tokens `acr` y los identificadores de identificación y `amr` afirma tener acceso a ellos. Su aplicación inspecciona el token de acceso o el token de identificación `acr` y las `amr` reclamaciones que contienen para evaluar el nivel de autenticación. Cuando un usuario actualiza sus tokens, Amazon Cognito conserva los `acr` mismos `amr` y los valores de los nuevos tokens de acceso e ID, por lo que no se le pide al usuario que vuelva a dar un paso adelante después de cada actualización. Para obtener más información sobre el contenido de los tokens, consulte Uso del token de ID y Uso del token de acceso.

El siguiente ejemplo muestra un token de ID después de que un usuario se autentique con una contraseña y un TOTP.

```
{
  "sub": "a1b2c3d4-5678-90ab-cdef-1234567890ab",
  "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_ExAmPlE",
  "aud": "abc123def456",
  "exp": 1759348800,
  "iat": 1759345200,
  "auth_time": 1759345200,
  "token_use": "id",
  "acr": "urn:cognito:loa:4",
  "amr": ["pwd", "otp", "mfa"],
  "email": "user@example.com",
  "cognito:username": "a1b2c3d4-5678-90ab-cdef-1234567890ab"
}
```

Cuando un usuario actualiza sus tokens, Amazon Cognito establece los `acr` mismos `amr` valores en los nuevos tokens de acceso e ID. Los valores se transfieren sin modificaciones. Este comportamiento proporciona una experiencia más fluida, ya que no se pide a los usuarios que vuelvan a dar un paso adelante después de cada actualización.

Actualizar un token no restablece la `auth_time` reclamación. Si quieres obligar a un usuario a volver a autenticarse después de un período de tiempo, usa el `max_age` parámetro, que compara la hora actual con `auth_time` independencia del número de veces que el usuario haya actualizado sus tokens.

Cuando un usuario completa la autenticación escalonada, Amazon Cognito emite nuevos tokens de acceso, ID y actualización con los registros `acr` actualizados `amr` y un nuevo período de validez.

Configuración de los nombres de los niveles de ACR

Puede personalizar el nombre URI de cada nivel de autenticación para su grupo de usuarios con el `AcrConfiguration` campo. Configure este campo al crear o actualizar un grupo de usuarios con la UpdateUserPooloperación CreateUserPoolo. Las combinaciones de factores que satisfacen cada nivel permanecen fijas. Solo personalizas los nombres.

La configuración de los nombres de los niveles de ACR personalizados con el `AcrConfiguration` campo requiere el plan de funciones Essentials o Plus. Para obtener más información, consulte Requisitos del plan de funciones.

El `AcrConfiguration` campo es un mapa con las teclas pulsadas`Level1`. `Level4` Cada valor es un objeto con una sola `AcrValue` cadena. Solo puede anular los niveles que desee personalizar. El siguiente ejemplo de solicitud personaliza los nombres de los niveles 2 y 3.

```
{
  "AcrConfiguration": {
    "Level2": { "AcrValue": "urn:my-standard:loa:2" },
    "Level3": { "AcrValue": "urn:my-standard:loa:3" }
  }
}
```

La respuesta siempre devuelve la configuración efectiva, con los nombres predeterminados combinados para los niveles que no haya personalizado.

```
{
  "AcrConfiguration": {
    "Level1": { "AcrValue": "urn:cognito:loa:1" },
    "Level2": { "AcrValue": "urn:my-standard:loa:2" },
    "Level3": { "AcrValue": "urn:my-standard:loa:3" },
    "Level4": { "AcrValue": "urn:cognito:loa:4" }
  }
}
```

Las siguientes reglas de validación se aplican a cada uno de ellos`AcrValue`.

- Cada uno `AcrValue` debe ser único en los cuatro niveles. Esto incluye los nombres predeterminados que se utilizan para rellenar los niveles que no se personalizan. Se rechaza un valor que choca con el valor predeterminado de un nivel no personalizado.
- Cada uno `AcrValue` es una cadena de 1 a 64 caracteres.
- Cada una no `AcrValue` puede contener espacios, comillas dobles ni barras invertidas.

Si cambias los nombres de los niveles de ACR, es posible que un usuario presente más adelante un token de acceso cuyo `acr` valor ya no coincide con ningún nivel configurado. Amazon Cognito no respeta el valor obsoleto y lo trata como un nivel no reconocido, pero no se trata de un error. Amazon Cognito sigue respetando los `amr` valores del token y solo solicita factores que aún no figuran en la `amr` reclamación, sin acreditar ningún nivel en particular.

Requisitos del plan de funciones

Las siguientes funciones requieren el plan de funciones Essentials o Plus.

- Solicitar una autenticación escalonada. Esto se aplica al `TARGET_ACR_VALUES` parámetro activado `InitiateAuth` y al `acr_values` parámetro del inicio de sesión gestionado.
- Configurar nombres de niveles de ACR personalizados con el `AcrConfiguration` campo activado `CreateUserPool` o`UpdateUserPool`.

El comportamiento al solicitar una autenticación intensificada con un plan de funciones insuficiente depende del punto de entrada.

- La `InitiateAuth` API devuelve un`FeatureUnavailableInTierException`.
- El punto final de autorización de inicio de sesión gestionado ignora silenciosamente `acr_values` y continúa con la autenticación normal, sin ningún error.

Para obtener más información sobre los planes de funciones, consulte. Planes de características de grupo de usuarios

Sus responsabilidades a la hora de aumentar la autenticación

Amazon Cognito emite tokens con el nivel de autenticación alcanzado por el usuario. Su aplicación es responsable de lo siguiente.

- Comprueba el identificador actual del usuario antes de redirigirlo. `auth_time`Evalúe su `max_age` nivel y el `acr` nivel del token de identificación con el nivel que requiere la operación, con los tokens que ya posee su aplicación. Redirija al usuario a Amazon Cognito para ampliarlo solo si aún no se ha alcanzado el nivel o la frescura requeridos.
- Establezca el `TARGET_ACR_VALUES` o correcto `acr_values` y el o correcto`MAX_AGE`. `max_age` Amazon Cognito actúa únicamente en función de lo que usted proporciona. No deduce el nivel o la frescura que pretende. Si solicita un nivel incorrecto u omite los parámetros, Amazon Cognito no mejora el nivel de usuario como pretendía.
- Valide la `acr` reclamación en su solicitud. Una vez que Amazon Cognito devuelva los tokens, la aplicación debe comprobar que la `acr` reclamación coincide con el nivel que requiere la operación antes de conceder el acceso. Amazon Cognito emite el token con el nivel alcanzado. La aplicación es responsable de hacer cumplir que una operación requiera un nivel determinado.

- Traduzca los niveles de Amazon Cognito a otros estándares ajenos al token. Amazon Cognito calcula el flujo de autenticación `acr` y las `amr` reclamaciones a partir del mismo. No puede personalizar, anular, añadir o suprimir estas notificaciones con un activador Lambda previo a la generación del token ni con ningún otro activador de Lambda. Si necesita traducir los niveles de Amazon Cognito a otro estándar, como NIST o eIDAS, hágalo en su capa de aplicación.

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 la protección contra amenazas 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 configurar la aplicación con un flujo de autenticación dl dispositivo de confianza. En este flujo, la aplicación puede presentar a los usuarios la opción de iniciar sesión sin MFA hasta más tarde, según determinen los requisitos de seguridad de la aplicación o las preferencias de los usuarios. Al final de ese período, la aplicación debe cambiar el estado del dispositivo a no recordado, y el usuario debe iniciar sesión con MFA hasta que confirme que quiere recordar un dispositivo. La aplicación puede, por ejemplo, solicitar a los usuarios que confíen en un dispositivo durante 30, 60 o 90 días. Puede almacenar esta fecha

en un atributo personalizado y, en dicha fecha, cambiar el estado recordado del dispositivo. A continuación, debe volver a solicitar al usuario que envíe un código de MFA y configurar el dispositivo para que se recuerde nuevamente después de una autenticación realizada correctamente.

1. 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 el menú 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 los dispositivos recordados para su grupo de usuarios en una solicitud de UpdateUserPoolAPI CreateUserPoolo una solicitud. Para obtener más información, consulte la DeviceConfigurationpropiedad.

La API de grupos de usuarios de Amazon Cognito tiene operaciones adicionales para recordar dispositivos.

1. ListDevicesy AdminListDevicesdevuelve una lista de las claves del dispositivo y sus metadatos para un usuario.
2. GetDevicey AdminGetDevicedevuelve la clave del dispositivo y los metadatos de un solo dispositivo.
3. UpdateDeviceStatusy AdminUpdateDeviceStatusconfigura el dispositivo de un usuario como recordado o no recordado.
4. ForgetDevicey AdminForgetDeviceeliminar 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 Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

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 ConfirmDevicesolicitud a 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 del usuario en una UpdateDeviceStatussolicitud.

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 la protección contra amenazas 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. Inicia la sesión de inicio de sesión de tu usuario con una solicitud de InitiateAuthAPI.
2. Responda a todos los desafíos de autenticación RespondToAuthChallengehasta que reciba los tokens web JSON (JWTs) que indican 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 para proporcionar SDKs 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 de su usuario en una solicitud de ConfirmDeviceAPI. 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 afirman que quieren recordar el dispositivo, genera una solicitud de UpdateDeviceStatusAPI con el token de acceso, la clave del dispositivo y `"DeviceRememberedStatus": "remembered"` el identificador del usuario.
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 de dispositivos en un flujo que utiliza activadores Lambda de desafío de autenticación personalizados, transfiere un `DEVICE_KEY` parámetro en tu solicitud de InitiateAuthAPI. 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. Inicie la sesión de inicio de sesión de su usuario con una solicitud de InitiateAuthAPI. 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 RespondToAuthChallengeAPI, envía un `ChallengeName` de `DEVICE_SRP_AUTH` y los parámetros para `USERNAMEDEVICE_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. Completa desafíos adicionales hasta que recibas los del usuario JWTs.

El siguiente pseudocódigo muestra cómo calcular los valores para la respuesta al desafío `DEVICE_PASSWORD_VERIFIER`. Para autenticar el SRP con un dispositivo, genera un nuevo secreto SRP para el usuario: una contraseña nueva de alta entropía `DeviceSecret`, una sal y el verificador de contraseñas asociado. Estos valores son distintos de la contraseña, la sal y el verificador utilizados para la autenticación SRP del usuario. Solo se utilizan para la autenticación

del dispositivo y solo se almacenan en el dispositivo. Las funciones para generar los secretos de SRP para los dispositivos de los usuarios están disponibles en las bibliotecas de SRP, que están disponibles en varias bibliotecas. SDKs

```
PASSWORD_CLAIM_SECRET_BLOCK = SECRET_BLOCK
TIMESTAMP = "Tue May 7 00:09:40 UTC 2025"
k = SHA256(N || g) as a non-negative integer in big-endian
u = SHA256(SRP_A || SRP_B) as a non-negative integer in big-endian
x = SHA256(salt || SHA256(DeviceGroupKey || DeviceKey || ":" || DeviceSecret)) as a
 non-negative integer in big-endian
S_USER = (SRP_B - k * g^x)^(a + u * x) % N
K_USER = HKDF_HMAC_SHA256(salt=u, ikm=S_USER, info="Caldera Derived Key", length=16
 bytes)
PASSWORD_CLAIM_SIGNATURE = Base64(HMAC_SHA256(key=K_USER, message=(DeviceGroupKey ||
 DeviceKey || PASSWORD_CLAIM_SECRET_BLOCK || TIMESTAMP)))
```

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. IdPs Los terceros deben gestionar por separado los dispositivos y el MFA de los usuarios que se autentiquen en Amazon Cognito. En el inicio de sesión administrado, puede solicitar el ámbito `aws.cognito.signin.user.admin`, pero el inicio de sesión administrado 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 de su usuario en una aplicación de cliente público, envíe la clave de acceso del usuario y la clave del dispositivo 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 AdminGetDeviceAPI con AWS credenciales y envíe el nombre de usuario, la clave del dispositivo y el grupo de usuarios de su 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, envía el token de acceso de tu usuario en una ListDevicessolicitud de API.
- En una aplicación de cliente confidencial, firma una solicitud de AdminListDevicesAPI con AWS credenciales y envía el nombre de usuario y el grupo de usuarios de tu 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, envía la clave de dispositivo y el token de acceso de tu usuario en ForgetDevicela solicitud de API.
- En una aplicación de cliente confidencial, envía la clave de dispositivo y el token de acceso de tu usuario en AdminForgetDeviceuna solicitud de API.

Seguridad avanzada con protección contra amenazas

Después de crear el grupo de usuarios, recibirá acceso a la opción Protección contra amenazas de la barra de navegación de la consola de Amazon Cognito. Puede activar las características de protección contra amenazas y personalizar las acciones que se llevan a cabo en respuesta a diferentes 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, Threat Protection publica las métricas en Amazon CloudWatch. Puede ver las métricas después de

que Amazon Cognito genere su primer evento. Consulte Visualización de las métricas de protección contra amenazas.

La protección contra amenazas, anteriormente conocida como características avanzadas de seguridad, consiste en un conjunto de herramientas de supervisión de la actividad no deseada en el grupo de usuarios y las herramientas de configuración para detener automáticamente cualquier actividad potencialmente maliciosa. La protección contra amenazas tiene diferentes opciones de configuración para las operaciones de autenticación estándar y personalizadas. Por ejemplo, puede que quiera enviar una notificación a un usuario con un inicio de sesión de autenticación personalizado sospechoso, en el que haya configurado factores de seguridad adicionales, pero también bloquear a un usuario con el mismo nivel de riesgo con una autenticación básica de nombre de usuario y contraseña.

La protección contra amenazas está disponible en el plan de características Plus. Para obtener más información, consulte Planes de características de grupo de usuarios.

Las siguientes opciones de grupo de usuarios son los componentes de la protección contra amenazas.

Credenciales comprometidas

Los usuarios reutilizan las contraseñas de varias cuentas de usuario. La función 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 si hay contraseñas que se suelen adivinar. Puede buscar credenciales comprometidas en los flujos de autenticación estándar de nombre de usuario y contraseña de los grupos de usuarios. Amazon Cognito no detecta las credenciales comprometidas en la contraseña remota segura (SRP) ni en la autenticación personalizada.

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.

Más información

Uso de la detección de credenciales comprometidas

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. Puede supervisar la actividad de los usuarios y automatizar las respuestas relativas a los niveles de riesgo detectados en el nombre de usuario, la contraseña, la SRP y la autenticación personalizada.

Al activar la protección contra amenazas, 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.

Más información

Uso de la autenticación flexible

Lista de direcciones IP permitidas y denegadas

Con la protección contra amenazas 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.

Lo que debe saber sobre las IP-address listas permitidas y las listas bloqueadas

- Debe expresar Bloquear siempre y Permitir siempre en formato CIDR, como `192.0.2.0/24`, una máscara de 24 bits o `192.0.2.252/32`, una sola dirección IP.
- Los dispositivos con direcciones IP incluidas en un rango de IP de bloqueo permanente no pueden registrarse ni iniciar sesión con aplicaciones de inicio de sesión gestionadas SDK-based ni con aplicaciones de inicio de sesión gestionadas, pero sí pueden iniciar sesión con aplicaciones de terceros. IdPs
- Las listas de Permitir siempre y Bloquear siempre no afectan a la actualización de los tokens.
- Amazon Cognito no aplica reglas de MFA de autenticación flexible a los dispositivos de un rango de direcciones IP Permitir siempre, pero sí aplica reglas de credenciales comprometidas.

Exportación de registros

La protección contra amenazas registra detalles pormenorizados de las solicitudes de autenticación de los usuarios en el grupo de usuarios. Estos registros incluyen evaluaciones de amenazas, información del usuario y metadatos de las sesiones, como la ubicación y el dispositivo. Para conservar y analizar estos registros, puede crear con ellos archivos externos. Los grupos de usuarios de Amazon Cognito exportan los registros de protección contra amenazas a Amazon S3, CloudWatch Logs y Amazon Data Firehose. Para obtener más información, consulte Visualización y exportación del historial de eventos de los usuarios.

Más información

Exportación de los registros de actividad de los usuarios en la protección contra amenazas

Temas

- Consideraciones y limitaciones en la protección contra amenazas
- Activación de la protección contra amenazas en los grupos de usuarios
- Conceptos de aplicación de la protección contra amenazas
- Protección contra amenazas para la autenticación estándar y la autenticación personalizada
- Requisitos previos de la protección contra amenazas
- Configuración de la protección contra amenazas
- Uso de la detección de credenciales comprometidas
- Uso de la autenticación flexible
- Recopilación de datos para la protección contra amenazas en las aplicaciones

Consideraciones y limitaciones en la protección contra amenazas

Las opciones de protección contra amenazas varían según los flujos de autenticación

Amazon Cognito admite tanto la autenticación flexible como la detección de credenciales comprometidas con los flujos de autenticación `USER_PASSWORD_AUTH` y `ADMIN_USER_PASSWORD_AUTH`. Solo puede habilitar la autenticación flexible para `USER_SRP_AUTH`. No puede utilizar la protección contra amenazas con el inicio de sesión federado.

Always-block Las IP contribuyen a solicitar cuotas

Las solicitudes bloqueadas de direcciones IP que figuran en una lista de excepciones Bloquear siempre del grupo de usuarios contribuye a las cuotas de tasas de solicitudes para los grupos de usuarios.

La protección contra amenazas no aplica límites de tasas

Existe un tipo de tráfico malicioso que se caracteriza por el gran volumen de solicitudes que genera; este es el caso de los ataques de denegación de servicio distribuido (DDoS). Las clasificaciones de riesgo que Amazon Cognito aplica al tráfico entrante se basan en la solicitud y no tienen en cuenta el volumen de solicitudes. Las solicitudes individuales en un evento de gran volumen pueden recibir una puntuación de riesgo y una respuesta automática por motivos relacionados con la capa de aplicaciones y no por su papel en un ataque volumétrico. Para implementar defensas contra los ataques volumétricos en sus grupos de usuarios, añada AWS WAF ACL web. Para obtener más información, consulte Asocie un AWS WAF ACL web a un grupo de usuarios.

La protección contra amenazas no afecta a las solicitudes de M2M

Las concesiones de credenciales de cliente estás pensadas para la autorización de máquina a máquina (M2M) sin conexión a las cuentas de usuario. La protección contra amenazas solo supervisa las cuentas de usuario y contraseñas de su grupo de usuarios. Para implementar funciones de seguridad en su actividad de M2M, tenga en cuenta las capacidades de monitoreo de las tasas de AWS WAF solicitudes y el contenido. Para obtener más información, consulte Asocie un AWS WAF ACL web a un grupo de usuarios.

Activación de la protección contra amenazas en los grupos de usuarios

Amazon Cognito user pools console

Cómo activar la protección contra amenazas 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. Si aún no lo ha hecho, active el plan de características Plus desde el menú Configuración.
5. Seleccione el menú Protección contra amenazas y Activar.
6. Seleccione Save changes (Guardar cambios).

API

Configura tu plan de funciones como Plus en una solicitud de UpdateUserPoolAPI CreateUserPoolo en una solicitud. El siguiente ejemplo parcial del cuerpo de la solicitud establece la protección contra amenazas en el modo de función completa. Para ver un ejemplo de solicitud completo, consulte Ejemplos.

```
"UserPoolAddOns": {
      "AdvancedSecurityMode": "ENFORCED"
   }
```

La protección contra amenazas es el término colectivo que designa a las características que supervisan las operaciones de los usuarios en busca de indicios de apropiación de cuentas y responden automáticamente para proteger las cuentas de usuario afectadas. Puede aplicar la configuración de protección contra amenazas a los usuarios cuando inician sesión con flujos de autenticación estándar y personalizados.

La protección contra amenazas genera registros que detallan el inicio y el cierre de sesión de los usuarios y otras actividades. Puede exportar estos registros a un sistema de terceros. Para obtener más información, consulte Visualización y exportación del historial de eventos de los usuarios.

Conceptos de aplicación de la protección contra amenazas

La protección contra amenazas comienza en un modo de solo auditoría, en el que el grupo de usuarios supervisa la actividad de los usuarios, asigna los niveles de riesgo y genera registros. Como práctica recomendada, ejecútalo en modo de solo auditoría durante dos semanas o más antes de activar el modo de función completa. Full-function este modo incluye un conjunto de reacciones automáticas ante la detección de actividades riesgosas y de contraseñas comprometidas. Con el modo de solo auditoría, puede supervisar las evaluaciones de amenazas que realiza Amazon Cognito. También puede proporcionar comentarios que servirán para entrenar la característica sobre falsos positivos y negativos.

Puede configurar la aplicación de la protección contra amenazas en el nivel del grupo de usuarios para que abarque a todos los clientes de aplicaciones del grupo de usuarios y en el nivel de los clientes de aplicaciones individuales. Las configuraciones de protección contra amenazas de los clientes de aplicación anulan la configuración del grupo de usuarios. Para configurar la protección contra amenazas de un cliente de aplicación, en la consola de Amazon Cognito, en el menú Clientes de aplicación de su grupo de usuarios, vaya a la configuración del cliente de aplicación. Allí, en

Utilizar configuración de nivel de cliente, configure la aplicación de la protección exclusivamente para el cliente de aplicación.

Además, puede configurar la protección contra amenazas por separado para los tipos de autenticación estándar y personalizada.

Protección contra amenazas para la autenticación estándar y la autenticación personalizada

Las formas de configuración de la protección contra amenazas dependen del tipo de autenticación que realice en el grupo de usuarios y en los clientes de aplicaciones. Cada uno de los siguientes tipos de autenticación puede tener su propio modo de aplicación y respuestas automáticas.

Autenticación estándar

La autenticación estándar consiste en el inicio de sesión, el cierre de sesión y la administración de contraseñas de los usuarios con flujos de nombre de usuario y contraseña, y en el inicio de sesión administrado. La protección contra amenazas de Amazon Cognito supervisa las operaciones en busca de indicadores de riesgo cuando se inicia sesión con el inicio de sesión administrado o se utilizan los siguientes parámetros de la API `AuthFlow`:

InitiateAuth

`USER_PASSWORD_AUTH`, `USER_SRP_AUTH`. La característica de credenciales comprometidas no tiene acceso a las contraseñas en inicio de sesión con `USER_SRP_AUTH` y no supervisa los eventos relacionados con este flujo ni actúa en consecuencia.

AdminInitiateAuth

`ADMIN_USER_PASSWORD_AUTH`, `USER_SRP_AUTH`. La característica de credenciales comprometidas no tiene acceso a las contraseñas en inicio de sesión con `USER_SRP_AUTH` y no supervisa los eventos relacionados con este flujo ni actúa en consecuencia.

Puede configurar el Modo de cumplimiento de la autenticación estándar en las opciones Solo auditoría o Función completa. Para deshabilitar la supervisión de amenazas para la autenticación estándar, establezca la protección contra amenazas en Sin aplicación.

Autenticación personalizada

La autenticación personalizada consiste en el inicio de sesión del usuario con desencadenadores de Lambda de desafío personalizados. No puede realizar una autenticación personalizada en

el inicio de sesión administrado. La protección contra amenazas de Amazon Cognito supervisa las operaciones en busca de indicadores de riesgo cuando se inicia sesión con el parámetro `AuthFlow` de la API `CUSTOM_AUTH` de `InitiateAuth` y `AdminInitiateAuth`.

Puede configurar el Modo de cumplimiento de la autenticación personalizada en las opciones Solo auditoría, Función completa o Sin aplicación. La opción Sin aplicación desactiva la supervisión de amenazas en la autenticación personalizada sin que afecte a las características de protección contra amenazas.

Requisitos previos de la protección contra amenazas

Antes de comenzar, necesitará lo siguiente:

- Un grupo de usuarios con un cliente de aplicación. Para obtener más información, consulte Creación de una nueva aplicación en la consola de Amazon Cognito.
- 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 la protección contra amenazas

Siga estas instrucciones para configurar la protección contra amenazas del grupo de usuarios.

> **Note**
>
> Para establecer una configuración de protección contra amenazas diferente para un cliente de aplicación en la consola de grupos de usuarios de Amazon Cognito, seleccione el cliente de aplicación en el menú Clientes de aplicación y elija Utilizar configuración de nivel de cliente.

Consola de administración de AWS

Cómo configurar la protección contra amenazas 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. Seleccione el menú Protección contra amenazas y Activar.
5. Elija el método de protección contra amenazas que desee configurar: Autenticación estándar y personalizada. Puede configurar diferentes modos de aplicación para la autenticación personalizada y estándar, pero ambas comparten la configuración de las respuestas automatizadas en el modo Función completa.
6. Seleccione Editar.
7. Elija un Modo de aplicación. Para empezar a responder inmediatamente a los riesgos detectados, seleccione Función completa y configure las respuestas automatizadas para las credenciales en riesgo y la autenticación flexible. Para recopilar información en los registros a nivel de usuario y en ellos CloudWatch, seleccione Solo auditoría.

   Recomendamos que mantenga la protección contra amenazas en el modo de auditoría durante dos semanas antes de activar las acciones. Durante este tiempo, Amazon Cognito puede aprender los patrones de uso de los usuarios de la aplicación y, por su parte, puede enviar comentarios sobre eventos para ajustar las respuestas.
8. Si seleccionó Audit only (Solo auditoría), elija Save changes (Guardar los cambios). Si seleccionó Full function (Función completa):

   a. Seleccione si desea realizar una acción personalizada o utilizar los valores predeterminados de Cognito para responder a las credenciales sospechosas de estar comprometidas. Los Cognito defaults (Valores predeterminados de Cognito) son:

      i. Detecte las credenciales comprometidas y el Sign-inSign-upcambio de contraseña.

      ii. Responder ante credenciales atacadas con la acción Block sign-in (Bloquear inicio de sesión).

   b. Si seleccionó Acciones personalizadas para las credenciales comprometidas, elija las acciones del grupo de usuarios que Amazon Cognito utiliza para la detección de eventos y las respuestas de credenciales comprometidas que desea que tome Amazon Cognito.

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 desea realizar una acción personalizada o utilizar los valores predeterminados de Cognito para responder a una supuesta actividad maliciosa. 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 ha seleccionado Acciones personalizadas para la autenticación adaptativa, elija las acciones de respuesta automática al riesgo que Amazon Cognito adopta en respuesta a los riesgos detectados en función del nivel de gravedad. Al asignar una respuesta a un nivel de riesgo, no 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. MFA opcional: si el usuario ha configurado la MFA, Amazon Cognito siempre requiere que el usuario proporcione un SMS adicional o un factor de contraseña de un solo uso basado en el tiempo (TOTP) al iniciar sesión. Si el usuario no tiene MFA configurada, puede seguir iniciando sesión normalmente.

   iii. Requerir MFA: si el usuario ha configurado la MFA, Amazon Cognito siempre exige que el usuario proporcione un SMS adicional o un factor TOTP al iniciar sesión. Si el usuario no tiene el MFA configurado, Amazon Cognito le pide que configure el 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.

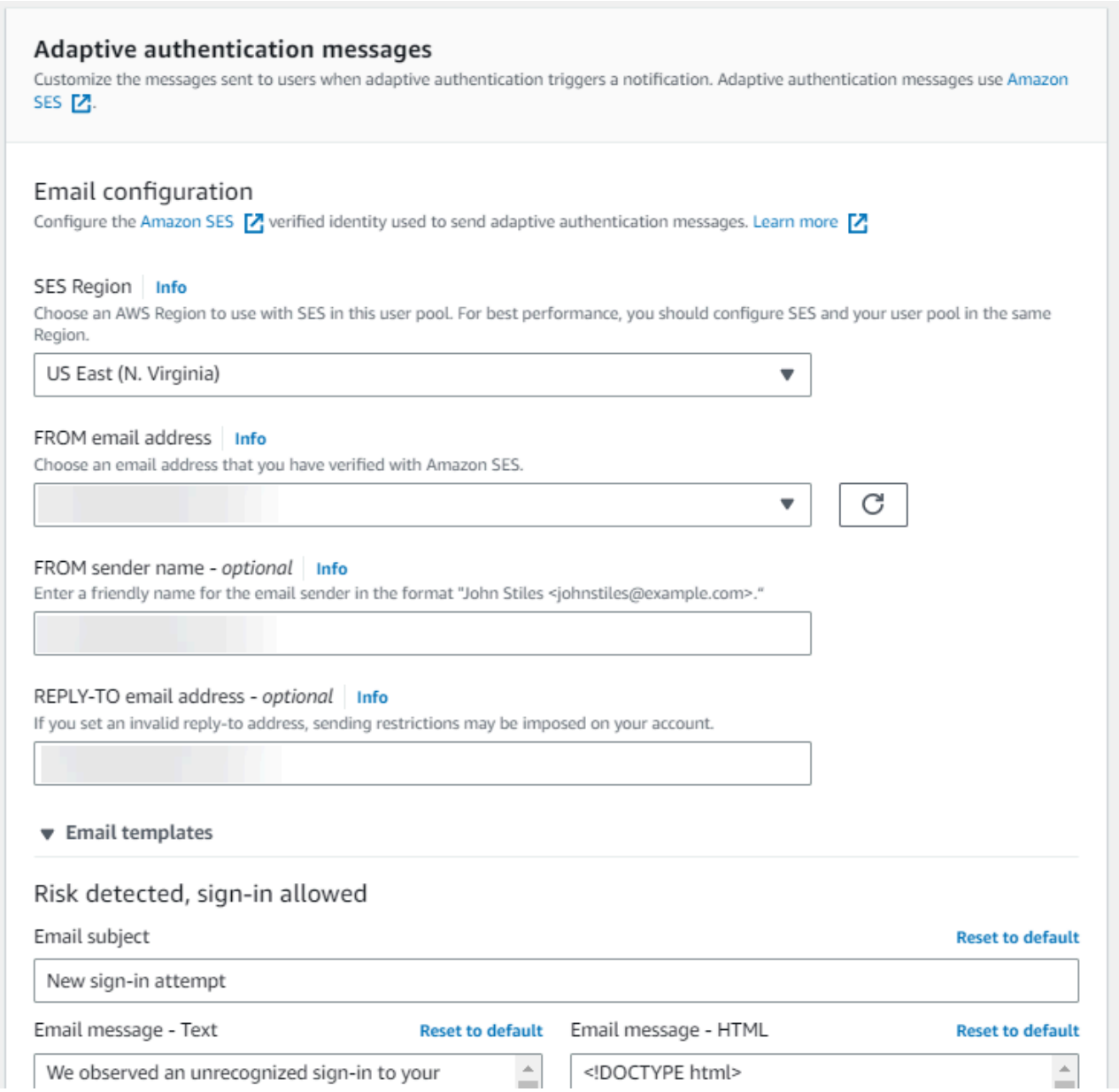
   iv. Block sign-in (Bloquear inicio de sesión): impide que el usuario inicie sesión.

   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.

9. 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 la configuración del correo electrónico, elija la región SES, la dirección de correo electrónico FROM, el nombre del remitente y la dirección de REPLY-TO correo

electrónico que desee utilizar con la autenticación adaptativa. 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.

10. Amplíe las excepciones de direcciones IP para crear una Always-allowo una Always-blocklista de rangos de direcciones IPv4 o IPv6 que siempre estén permitidos o bloqueados,

independientemente de la evaluación de riesgos de protección contra amenazas. Especifique los rangos de direcciones IP en notación CIDR (por ejemplo, 192.168.100). 0/24).

11. Seleccione Save changes (Guardar cambios).

API (user pool)

Para establecer la configuración de protección contra amenazas para un grupo de usuarios, envíe una solicitud de SetRiskConfigurationAPI que incluya un `UserPoolId` parámetro, pero no `ClientId` uno. A continuación, se muestra un ejemplo de cuerpo de solicitud de un grupo de usuarios. Esta configuración de riesgo ejecuta una serie de acciones cada vez mayores en función de la gravedad del riesgo y envía notificaciones a los usuarios en todos los niveles de riesgo. Aplica un bloqueo de credenciales comprometidas a las operaciones de registro.

Para aplicar esta configuración, debe `AdvancedSecurityMode` configurarla `ENFORCED` en una solicitud independiente CreateUserPoolo de UpdateUserPoolAPI. Para obtener más información sobre las plantillas de marcadores de posición, como `{username}` en este ejemplo, consulte Configuración de los mensajes de verificación, invitación, autenticación y MFA.

```
{
   "AccountTakeoverRiskConfiguration": {
      "Actions": {
         "HighAction": {
            "EventAction": "MFA_REQUIRED",
            "Notify": true
         },
         "LowAction": {
            "EventAction": "NO_ACTION",
            "Notify": true
         },
         "MediumAction": {
            "EventAction": "MFA_IF_CONFIGURED",
            "Notify": true
         }
      },
      "NotifyConfiguration": {
         "BlockEmail": {
            "Subject": "You have been blocked for suspicious activity",
            "TextBody": "We blocked {username} at {login-time} from {ip-address}."
         },
         "From": "admin@example.com",
         "MfaEmail": {
```

```
            "Subject": "Suspicious activity detected, MFA required",
            "TextBody": "Unexpected sign-in from {username} on device {device-name}.
 You must use MFA."
         },
         "NoActionEmail": {
            "Subject": "Suspicious activity detected, secure your user account",
            "TextBody": "We noticed suspicious sign-in activity by {username} from
 {city}, {country} at {login-time}. If this was not you, reset your password."
         },
         "ReplyTo": "admin@example.com",
         "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
      }
   },
   "CompromisedCredentialsRiskConfiguration": {
      "Actions": {
         "EventAction": "BLOCK"
      },
      "EventFilter": [ "SIGN_UP" ]
   },
   "RiskExceptionConfiguration": {
      "BlockedIPRangeList": [ "192.0.2.0/24","198.51.100.0/24" ],
      "SkippedIPRangeList": [ "203.0.113.0/24" ]
   },
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

API (app client)

Para establecer la configuración de protección contra amenazas para un cliente de aplicaciones, envía una solicitud de SetRiskConfigurationAPI que incluya un `UserPoolId` parámetro y un `ClientId` parámetro. A continuación, se muestra un ejemplo de cuerpo de solicitud de un cliente de aplicación. Esta configuración de riesgo es más estricta que la configuración del grupo de usuarios y bloquea las entradas de alto riesgo. También aplica bloques de credenciales comprometidas a las operaciones de registro, inicio de sesión y restablecimiento de contraseñas.

Para aplicar esta configuración, debes `AdvancedSecurityMode` configurarla `ENFORCED` en una solicitud independiente CreateUserPoolo de UpdateUserPoolAPI. Para obtener más información sobre las plantillas de marcadores de posición, como `{username}` en este ejemplo, consulte Configuración de los mensajes de verificación, invitación, autenticación y MFA.

```
{
```

```
   "AccountTakeoverRiskConfiguration": {
      "Actions": {
         "HighAction": {
            "EventAction": "BLOCK",
            "Notify": true
         },
         "LowAction": {
            "EventAction": "NO_ACTION",
            "Notify": true
         },
         "MediumAction": {
            "EventAction": "MFA_REQUIRED",
            "Notify": true
         }
      },
      "NotifyConfiguration": {
         "BlockEmail": {
            "Subject": "You have been blocked for suspicious activity",
            "TextBody": "We blocked {username} at {login-time} from {ip-address}."
         },
         "From": "admin@example.com",
         "MfaEmail": {
            "Subject": "Suspicious activity detected, MFA required",
            "TextBody": "Unexpected sign-in from {username} on device {device-name}.
 You must use MFA."
         },
         "NoActionEmail": {
            "Subject": "Suspicious activity detected, secure your user account",
            "TextBody": "We noticed suspicious sign-in activity by {username} from
 {city}, {country} at {login-time}. If this was not you, reset your password."
         },
         "ReplyTo": "admin@example.com",
         "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
      }
   },
   "ClientId": "1example23456789",
   "CompromisedCredentialsRiskConfiguration": {
      "Actions": {
         "EventAction": "BLOCK"
      },
      "EventFilter": [ "SIGN_UP", "SIGN_IN", "PASSWORD_CHANGE" ]
   },
   "RiskExceptionConfiguration": {
```

```
      "BlockedIPRangeList": [ "192.0.2.1/32","192.0.2.2/32" ],
      "SkippedIPRangeList": [ "192.0.2.3/32","192.0.2.4/32" ]
   },
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Uso de la detección de credenciales comprometidas

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 el inicio de sesión administrado y con la API de Amazon Cognito.

En la consola de Amazon Cognito, en el menú Protección contra amenazas, puede configurar 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 protección contra amenazas. 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.

Las credenciales comprometidas pueden comprobar las contraseñas de las siguientes actividades de usuario.

Sign-up

> Su grupo de usuarios comprueba las contraseñas que los usuarios transmiten durante la SignUpoperación y en la página de registro del inicio de sesión gestionado para ver si hay indicios de que están en peligro.

Sign-in

Su grupo de usuarios comprueba las contraseñas que los usuarios envían al iniciar sesión con contraseñas para detectar indicios de que están en peligro. Amazon Cognito puede revisar el `ADMIN_USER_PASSWORD_AUTH` flujo de entrada AdminInitiateAuth, el `USER_PASSWORD_AUTH` flujo de entrada InitiateAuthy la `PASSWORD` opción del `USER_AUTH` flujo en ambos.

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.

Restablecimiento de la contraseña

Con la operación de ConfirmForgotPasswordautoservicio de restablecimiento de contraseñas, su grupo de usuarios comprueba si hay indicios de riesgo en las operaciones que establecen contraseñas para nuevos usuarios. El código necesario para esta operación lo genera ForgotPasswordy AdminResetUserPassword.

Las credenciales comprometidas no comprueban las contraseñas temporales o permanentes establecidas por el administrador. AdminSetUserPassword Sin embargo, en el caso de las contraseñas temporales, el grupo de usuarios comprueba las contraseñas en función de las respuestas al `NEW_PASSWORD_REQUIRED` desafío en y. RespondToAuthChallengeAdminRespondToAuthChallenge

Para añadir protecciones contra credenciales atacadas a un grupo de usuarios, consulte Seguridad avanzada con protección contra amenazas.

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 los factores de dispositivo y usuario que su aplicación proporciona y otros factores que Amazon Cognito deduce de la solicitud. Algunos factores que contribuyen a la evaluación del riesgo por parte de Amazon Cognito son la dirección IP , el agente de usuario y la distancia geográfica con respecto a otros intentos de inicio de sesión.

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 métricas sobre los intentos de inicio de sesión, sus niveles de riesgo y las impugnaciones fallidas a Amazon. CloudWatch Para obtener más información, consulte Visualización de las métricas de protección contra amenazas.

Para añadir la autenticación flexible a un grupo de usuarios, consulte Seguridad avanzada con protección contra amenazas.

Temas

- Información general sobre la autenticación flexible
- Adición de datos de sesión y dispositivos de usuario a las solicitudes de API
- Visualización y exportación 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 consola de Amazon Cognito, en el menú Protección contra amenazas, puede elegir la configuración de la autenticación flexible, como las acciones que se deben realizar en los distintos niveles de riesgo y la personalización de los mensajes de notificación para los usuarios. Puede asignar una configuración de protección contra amenazas global a todos sus clientes de aplicación, 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.

Tenga en cuenta sus opciones detenidamente al cambiar su método de cumplimiento de a Audit-only. Full-function 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	Action
Permitir	Los usuarios pueden iniciar sesión sin un factor adicional.
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.
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 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 protección contra amenazas 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 Threat Protection 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 la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles. Puede enviar `ContextData` si tiene el plan de características Plus. Para obtener más información, consulte Configuración de la protección contra amenazas.

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 la protección contra amenazas 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:

- Su grupo de usuarios está incluido en el plan de características Plus. Para obtener más información, consulte Planes de características de grupo de usuarios.

- El cliente de aplicación tiene un secreto de cliente. Para obtener más información, consulte Application-specific configuración con clientes de aplicaciones.
- Ha activado Accept additional user context data (Aceptar datos de contexto de usuario adicionales) en el cliente de aplicación. Para obtener más información, consulte Aceptar datos contextuales de usuario adicionales (Consola de administración de AWS).

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

Aceptar datos contextuales de usuario adicionales (Consola de administración de AWS)

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 la protección contra amenazas 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. Seleccione el menú Clientes de aplicación.
4. Elija un cliente de aplicación o cree uno. 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. Seleccione Save changes (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 más información sobre cómo trabajar con la protección contra amenazas en la aplicación web o móvil, consulte Recopilación de datos para la protección contra amenazas en las aplicaciones. 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 protección contra amenazas 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: usuario ContextData
- JavaScript: amazon-cognito-advanced-security-data.min.js

Visualización y exportación del historial de eventos de los usuarios

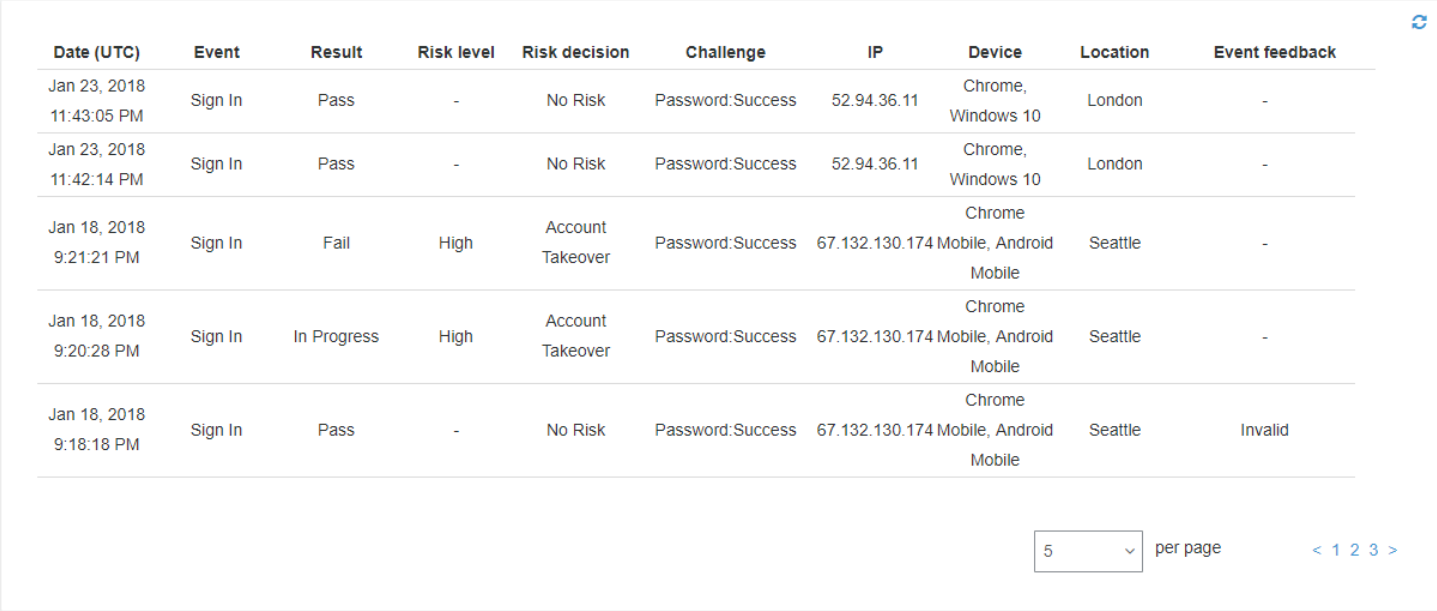
Amazon Cognito genera un registro por cada evento de autenticación realizado por un usuario cuando se habilita la protección contra amenazas. De forma predeterminada, puede ver los registros de los usuarios en el menú Usuarios de la consola de Amazon Cognito o mediante la operación de la AdminListUserAuthEventsAPI. También puede exportar estos eventos a un sistema externo como CloudWatch Logs, Amazon S3 o Amazon Data Firehose. La característica de exportación hace que la información de seguridad sobre la actividad de los usuarios en la aplicación sea más accesible a sus propios sistemas de análisis de seguridad.

Temas

- Ver el historial de eventos del usuario (Consola de administración de AWS)
- Visualización del historial de eventos de los usuarios (API/CLI)
- Exportación de eventos de autenticación de usuarios

Ver el historial de eventos del usuario (Consola de administración de AWS)

Para ver el historial de inicios de sesión de un usuario, puede elegir el usuario en el menú 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.

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 Descripción de los tokens web JSON (JWT) de los grupos de usuarios.

Visualización del historial de eventos de los usuarios (API/CLI)

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.

AdminListUserAuthEvents request

El siguiente cuerpo de la solicitud de `AdminListUserAuthEvents` devuelve el registro de actividad más reciente de un usuario.

```
{
  "UserPoolId": "us-west-2_EXAMPLE",
  "Username": "myexampleuser",
  "MaxResults": 1
}
```

admin-list-user-auth-events request

La siguiente solicitud de `admin-list-user-auth-events` devuelve el registro de actividad más reciente de un usuario.

```
aws cognito-idp admin-list-user-auth-events --max-results 1 --username myexampleuser
 --user-pool-id us-west-2_EXAMPLE
```

Response

Amazon Cognito devuelve el mismo cuerpo de respuesta JSON a ambas solicitudes. El siguiente es un ejemplo de respuesta para un evento de inicio de sesión en el inicio de sesión administrado que no contenía factores de riesgo:

```
{
    "AuthEvents": [
```

```
        {
            "EventId": "[event ID]",
            "EventType": "SignIn",
            "CreationDate": "[Timestamp]",
            "EventResponse": "Pass",
            "EventRisk": {
                "RiskDecision": "NoRisk",
                "CompromisedCredentialsDetected": false
            },
            "ChallengeResponses": [
                {
                    "ChallengeName": "Password",
                    "ChallengeResponse": "Success"
                }
            ],
            "EventContextData": {
                "IpAddress": "192.168.2.1",
                "DeviceName": "Chrome 125, Windows 10",
                "Timezone": "-07:00",
                "City": "Bellevue",
                "Country": "United States"
            }
        }
    ],
    "NextToken": "[event ID]#[Timestamp]"
}
```

Exportación de eventos de autenticación de usuarios

Configure el grupo de usuarios para exportar eventos de los usuarios desde la protección contra amenazas hasta un sistema externo. Los sistemas externos compatibles (Amazon S3, CloudWatch Logs y Amazon Data Firehose) pueden añadir costes a su AWS factura por los datos que envíe o recupere. Para obtener más información, consulte Exportación de los registros de actividad de los usuarios en la protección contra amenazas.

Consola de administración de AWS

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. Seleccione el menú Registrar la transmisión. Seleccione Editar.

5. En Estado de registro, seleccione la casilla de verificación situada junto a Activar la exportación del registro de actividad del usuario.
6. En Destino del registro, elige el Servicio de AWS que quieres que gestione tus registros: grupo de CloudWatch registros, transmisión de Amazon Data Firehose o bucket de S3.
7. Su selección rellena el selector de recursos con el tipo de recurso correspondiente. Seleccione un grupo de registro, un flujo o un bucket de la lista. También puede seleccionar el botón Crear Consola de administración de AWS para ir al servicio seleccionado y crear un recurso nuevo.
8. Seleccione Guardar cambios.

API

Elija un tipo de destino para los registros de actividad de los usuarios.

A continuación se muestra un ejemplo de cuerpo de solicitud `SetLogDeliveryConfiguration` que establece un flujo de Firehose como destino del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "FirehoseConfiguration": {
            "StreamArn": "arn:aws:firehose:us-west-2:123456789012:deliverystream/
example-user-pool-activity-exported"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

A continuación se muestra un ejemplo de cuerpo de solicitud `SetLogDeliveryConfiguration` que establece un bucket de Amazon S3 como destino del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "S3Configuration": {
            "BucketArn": "arn:aws:s3:::amzn-s3-demo-logging-bucket"
```

```
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

A continuación, se muestra un ejemplo del cuerpo de la `SetLogDeliveryConfiguration` solicitud que establece un grupo de CloudWatch registros como destino del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "CloudWatchLogsConfiguration": {
            "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-group:DOC-
EXAMPLE-LOG-GROUP"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

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 Usuarios y seleccione 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 los detalles del usuario en el menú Usuarios. Si selecciona una entrada, puede marcar el evento como válido o no válido. También puedes enviar comentarios mediante la operación AdminUpdateAuthEventFeedbackde la API del grupo de usuarios y mediante el AWS CLI comando admin-update-auth-event-feedback.

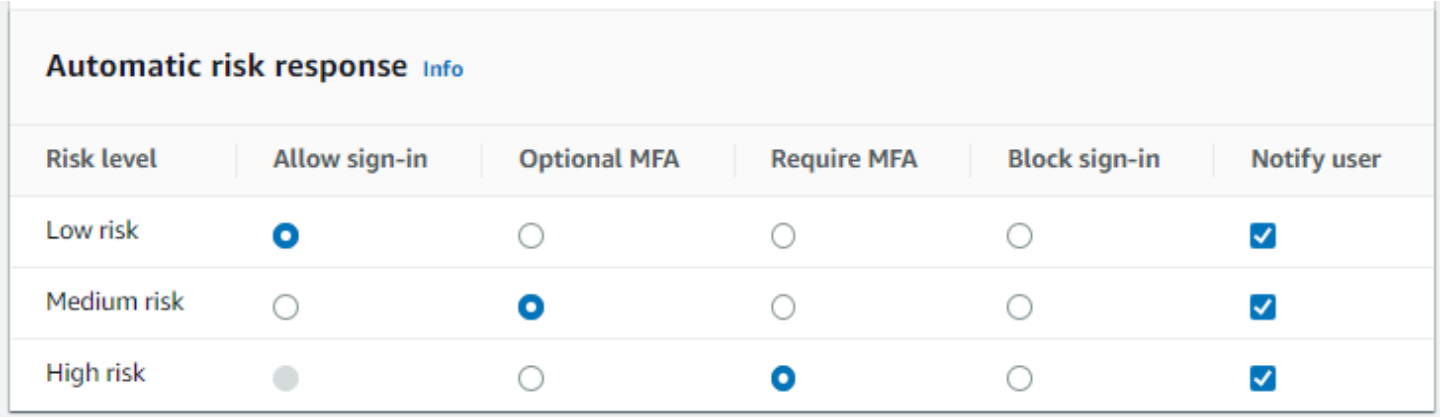
Al seleccionar Set as valid (Establecer como válido) en la consola de Amazon Cognito o proporcionar un valor `FeedbackValue` de `valid` en la API, le está indicando a Amazon Cognito que confía en una sesión de usuario en la que Amazon Cognito ha determinado que hay cierto nivel de riesgo. Al seleccionar 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 la protección contra amenazas, 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.

> **Note**
>
> Amazon Cognito solo envía mensajes de notificación a los usuarios cuando su acción genera una respuesta automática al riesgo: bloquear el inicio de sesión, permitir el inicio de sesión, configurar la MFA como opcional o requerir la MFA. Es posible que a algunas solicitudes se les asigne un nivel de riesgo, pero que no generen respuestas de riesgo automatizadas con autenticación flexible; en este caso, su grupo de usuarios no envía notificaciones. Por ejemplo, es posible que se registren contraseñas incorrectas con una clasificación de riesgo y que la respuesta de Amazon Cognito consista en no efectuar el inicio de sesión, no en aplicar una regla de autenticación flexible.

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 Plantillas de correo electrónico desde Mensajes de autenticación flexible en su configuración de protección contra amenazas. Para obtener más información sobre las plantillas de correo electrónico, consulte Plantillas de mensaje.

Recopilación de datos para la protección contra amenazas en las aplicaciones

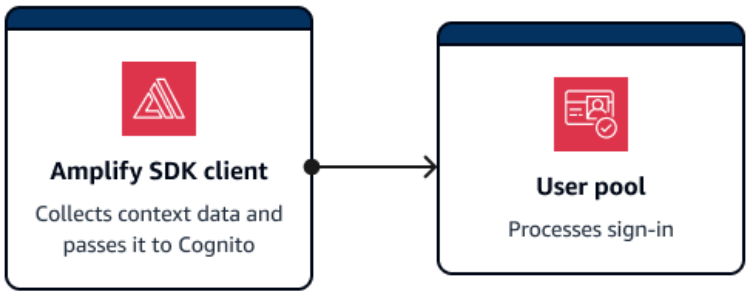
La autenticación flexible de Amazon Cognito evalúa los niveles de riesgo de los intentos de apropiación de cuentas a partir de los detalles contextuales de los intentos de inicio de sesión de los usuarios. La aplicación debe añadir datos de contexto a las solicitudes de la API para que la protección contra amenazas de Amazon Cognito pueda evaluar el riesgo con más precisión. Los datos de contexto son información como la dirección IP, el agente del navegador, la información del dispositivo y los encabezados de las solicitudes que proporcionan información contextual sobre cómo se ha conectado un usuario al grupo de usuarios.

La responsabilidad principal de una aplicación que envía este contexto a Amazon Cognito es un parámetro `EncodedData` en las solicitudes de autenticación a grupos de usuarios. Para añadir estos datos a sus solicitudes, puede implementar Amazon Cognito con un SDK que genere automáticamente esta información por usted, o puede implementar un módulo para JavaScript iOS o Android que recopile estos datos. Client-onlylas aplicaciones que realizan solicitudes directas a Amazon Cognito deben implementar AWS Amplify los SDK. Client-serverlas aplicaciones que tienen un servidor intermedio o un componente de API deben implementar un módulo de SDK independiente.

En las siguientes situaciones, la interfaz de autenticación administra la recopilación de datos del contexto del usuario sin ninguna configuración adicional:

- El inicio de sesión administrado recopila y envía automáticamente los datos de contexto al sistema de protección contra amenazas.
- Todas AWS Amplify las bibliotecas incorporan la recopilación de datos contextuales en sus métodos de autenticación.

Envío de datos de contexto de usuario en aplicaciones exclusivas para clientes con Amplify



Los SDK de Amplify son compatibles con los clientes móviles que se autentican directamente con Amazon Cognito. Los clientes de este tipo realizan solicitudes de API directas a las operaciones de API públicas de Amazon Cognito. Los clientes de Amplify recopilan automáticamente datos de contexto para protección contra amenazas de forma predeterminada.

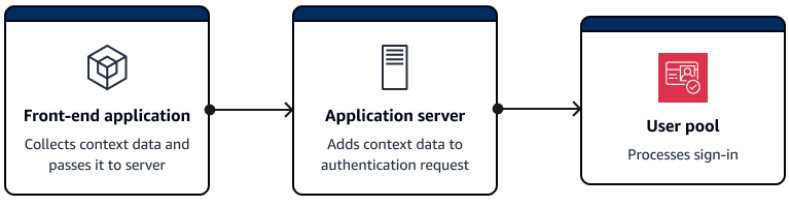
Amplify las aplicaciones con es una JavaScript excepción. Requieren la adición de un JavaScript módulo que recopile datos de contexto del usuario.

Por lo general, una aplicación en esta configuración utiliza operaciones de API no autenticadas, como InitiateAuthy. RespondToAuthChallenge El UserContextDataobjeto ayuda a evaluar los riesgos de estas operaciones con mayor precisión. Los SDK de Amplify añaden información sobre el dispositivo y la sesión a un parámetro `EncodedData` de `UserContextData`.

Recopilación de datos de contexto en aplicaciones cliente-servidor

Algunas aplicaciones tienen un nivel frontend que recopila los datos de autenticación de los usuarios y un nivel backend de aplicaciones que envía las solicitudes de autenticación a Amazon Cognito. Se trata de una arquitectura común en los servidores web y las aplicaciones respaldadas por

microservicios. En estas aplicaciones, debe importar una biblioteca pública de recopilación de datos contextuales.



Normalmente, un servidor de aplicaciones en esta configuración utiliza operaciones de API autenticadas, como y. AdminInitiateAuthAdminRespondToAuthChallenge El ContextDataobjeto ayuda a Amazon Cognito a evaluar los riesgos de estas operaciones con mayor precisión. El contenido de `ContextData` son los datos codificados que el frontend ha pasado al servidor y detalles adicionales de la solicitud HTTP del usuario al servidor. Estos detalles contextuales adicionales, como los encabezados HTTP y la dirección IP, proporcionan al servidor de aplicaciones las características del entorno del usuario.

Es posible que su servidor de aplicaciones también inicie sesión con operaciones de API no autenticadas, como y. InitiateAuthRespondToAuthChallenge El UserContextDataobjeto sirve de base para el análisis de riesgos de la protección contra amenazas en estas operaciones. Las operaciones de las bibliotecas de recopilación de datos de contexto público disponibles añaden información de seguridad al parámetro `EncodedData` en las solicitudes de autenticación. Además, debe configurar el grupo de usuarios para que acepte datos de contexto adicionales y añada la IP de origen del usuario al parámetro `IpAddress` de `UserContextData`.

Adición de datos de contexto en aplicaciones cliente-servidor

1. En su aplicación front-end, recopile datos de contexto codificados del cliente con un JavaScript módulo, iOS o Android.
2. Transfiera los datos codificados y los detalles de la solicitud de autenticación al servidor de aplicaciones.
3. En el servidor de aplicaciones, extraiga de la solicitud HTTP la dirección IP del usuario, los encabezados HTTP pertinentes, el nombre del servidor solicitado y la ruta solicitada. Rellene estos valores con el ContextDataparámetro de su solicitud de API a Amazon Cognito.
4. Rellene el parámetro `EncodedData` de `ContextData` en la solicitud de API con los datos codificados del dispositivo que el módulo del SKD ha recopilado. Añada estos datos de contexto a la solicitud de autenticación.

Bibliotecas de datos contextuales para aplicaciones cliente-servidor

JavaScript

El módulo `amazon-cognito-advanced-security-data.min.js` recopila los `EncodedData` que puede transferir al servidor de aplicaciones.

Añada el `amazon-cognito-advanced-security-data.min.js` módulo a su configuración. JavaScript `<region>`Sustitúyalo por uno Región de AWS de la siguiente lista: `us-east-1` `us-east-2``us-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>
```

Para generar un `encodedContextData` objeto que pueda usar en el `EncodedData` parámetro, añada lo siguiente a la fuente de JavaScript la aplicación:

```
var encodedContextData = AmazonCognitoAdvancedSecurityData.getData(_username,
 _userpoolId, _userPoolClientId);
```

iOS/Swift

Para generar datos de contexto, las aplicaciones iOS pueden integrar el módulo Mobile SDK for iOS AWSCognitoIdentityProviderASF.

A fin de recopilar datos contextuales codificados para la protección contra amenazas, añada el siguiente fragmento a la aplicación:

```
import AWSCognitoIdentityProviderASF

let deviceId = getDeviceId()
let encodedContextData = AWSCognitoIdentityProviderASF.userContextData(
                            userPoolId,
                            username: username,
                            deviceId: deviceId,
                            userPoolClientId: userPoolClientId)

/**
 * Reuse DeviceId from keychain or generate one for the first time.
 */
func getDeviceId() -> String {
```

```
    let deviceIdKey = getKeyChainKey(namespace: userPoolId, key:
 "AWSCognitoAuthAsfDeviceId")

   if let existingDeviceId = self.keychain.string(forKey: deviceIdKey) {
        return existingDeviceId
    }

    let newDeviceId = UUID().uuidString
    self.keychain.setString(newDeviceId, forKey: deviceIdKey)
    return newDeviceId
}

/**
 * Get a namespaced keychain key given a namespace and key
 */
func getKeyChainKey(namespace: String, key: String) -> String {
    return "\(namespace).\(key)"
}
```

Android

Para generar datos de contexto, las aplicaciones de Android pueden integrar el módulo aws-android-sdk-cognitoidentityprovider-asf de Mobile SDK para Android.

A fin de recopilar datos contextuales codificados para la protección contra amenazas, añada el siguiente fragmento a la aplicación:

```
UserContextDataProvider provider = UserContextDataProvider.getInstance();
// context here is android application context.
String encodedContextData = provider.getEncodedContextData(context, username,
 userPoolId, userPoolClientId);
```

Asocie un AWS WAF ACL 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 clásica, inicio de sesión gestionado y puntos de enlace del servicio de 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 acerca de ACL web de AWS WAF , consulte Administración y uso de una lista de control de acceso web (ACL web)en la Guía para desarrolladores de AWS WAF .

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. Los componentes de la solicitud a los que Amazon Cognito reenvía AWS WAF dependen del tipo de solicitud. En el caso del inicio de sesión gestionado y la clásica interfaz de usuario alojada, Amazon Cognito no reenvía el cuerpo de la solicitud. Para obtener más información, consulte Cosas que debe saber AWS WAF ACL web y Amazon Cognito.

Cosas que debe saber AWS WAF ACL web y Amazon Cognito

- No puede configurar las reglas de ACL web para que coincidan con la información de identificación personal (PII) de las solicitudes de grupos de usuarios, por ejemplo, nombres de usuario, contraseñas, números de teléfono o direcciones de correo electrónico. Estos datos no estarán disponibles para. AWS WAF En lugar de esto, configure las reglas de ACL web para que coincidan con los datos de sesión de los encabezados, la ruta y el cuerpo, como las direcciones IP, los agentes del navegador y las operaciones de API solicitadas.
- Amazon Cognito admite reglas de ACL web basadas en el cuerpo solo para solicitudes a la API de grupos de usuarios. Para el inicio de sesión gestionado y la clásica interfaz de usuario alojada, Amazon Cognito no reenvía el cuerpo de la solicitud y solo inspecciona los encabezados y otros componentes que no son del cuerpo.
- Las reglas de ACL web solo pueden devolver su respuesta de bloqueo personalizada cuando el navegador carga una página de inicio de sesión administrada. En su lugar, las solicitudes que la página envía sin volver a cargarse, como cuando un usuario envía el formulario de inicio de sesión, reciben una respuesta de `403` bloqueo predeterminada.
- 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 API-level limitación.
- 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 cada 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 aprender a configurar cómo gestionar las solicitudes AWS WAF 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. La característica ATP se encuentra en el grupo de reglas administradas `AWS-AWSManagedRulesATPRuleSet`. Antes de asociar una ACL web con un grupo de usuarios, asegúrese de que no utilice este grupo de reglas administradas.
- 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 al iniciar sesión gestionada en el registro de TP. Para crear una regla que tenga una acción CAPTCHA y no afecte al TOTP del inicio de sesión administrado, consulte Configurando su AWS WAF ACL web para inicio de sesión gestionado TOTP MFA.
- Cuando una regla de su ACL web devuelve una respuesta de bloqueo personalizada, Amazon Cognito añade un Content-Security-Policy encabezado a esa respuesta. Esta política solo permite scripts del mismo origen que el dominio de su grupo de usuarios y de los scripts de AWS WAF CAPTCHA y de desafío que proporciona Amazon Cognito. Bloquea cualquier otro script en línea o de terceros, incluido el script que su cuerpo de respuesta personalizado utilice para redirigir al usuario. Para obtener más información, consulte Content-Security-Policy en las respuestas de bloque personalizadas.
- Cuando una regla de su ACL web devuelve una respuesta de bloqueo personalizada con un código de estado HTTP igual o superior a 500 en respuesta a un inicio de sesión gestionado o a una solicitud de interfaz de usuario alojada clásica, Amazon Cognito devuelve un código de `403 (Forbidden)` estado en su lugar. Para obtener más información, consulte Los códigos de estado de las respuestas de bloqueo personalizadas.

AWS WAF inspecciona las solicitudes enviadas a los siguientes puntos de enlace.

El inicio de sesión administrado y la interfaz de usuario alojada clásica

Solicitudes a todos los puntos de conexión en Referencia de los puntos de conexión de grupos de usuarios y del inicio de sesión administrado.

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 Lista de operaciones de API agrupadas por modelo de autorización.

Puede configurar las reglas en la ACL web con acciones de reglas como Recuento, Permiso o Bloqueo o presentar un CAPTCHA en respuesta a una solicitud que coincida 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 al error dependen de la forma en que realices una solicitud a la API.
>
> - Puede personalizar el código de error y el cuerpo de respuesta de las solicitudes de inicio de sesión administrado. Amazon Cognito devuelve códigos de error personalizados de 500 o más como. `403` Para obtener más información, consulte Los códigos de estado de las respuestas de bloqueo personalizadas. Solo puede presentar un CAPTCHA para que el usuario lo resuelva en el inicio de sesión administrado.
> - 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.
> - El AWS Command Line Interface (AWS CLI) y el AWS SDK devuelven un `ForbiddenException` error a las solicitudes que generan una respuesta de bloqueo o CAPTCHA.

Content-Security-Policy en las respuestas de bloque personalizadas

Si tu respuesta de bloqueo personalizada se basa en scripts, esta política puede afectar a su comportamiento. Cuando una regla de ACL web devuelve una respuesta de bloqueo personalizada al inicio de sesión gestionado o a la interfaz de usuario alojada clásica, Amazon Cognito añade un encabezado Content-Security-Policy (CSP) a esa respuesta. El CSP es un mecanismo de seguridad del navegador que restringe los scripts y otros recursos que puede cargar una página. La política restringe la respuesta de la siguiente manera:

- La respuesta puede incluir scripts y otros recursos del mismo origen que el dominio del grupo de usuarios.
- La respuesta puede incluir el AWS WAF CAPTCHA y los scripts de desafío que proporciona Amazon Cognito. Amazon Cognito permite incluir estos scripts en listas para que las acciones de AWS WAF CAPTCHA y de las reglas de desafío sigan funcionando.

No se ejecuta ningún otro script en línea ni ningún script alojado en un dominio que no sea el dominio de su grupo de usuarios. Esto incluye un script que un cuerpo de respuesta personalizado utiliza para redirigir al usuario, enviar un formulario o enviar una solicitud de red.

Si tu respuesta personalizada necesita redirigir al usuario, devuelve un código de estado HTTP 3xx y un `Location` encabezado en lugar de una redirección basada en scripts. Este patrón no depende de la ejecución del script y no se ve afectado por el CSP.

Este comportamiento se aplica a todos los Regiones de AWS lugares en los que se puede asociar una ACL web a un grupo de usuarios.

Los códigos de estado de las respuestas de bloqueo personalizadas

Amazon Cognito no devuelve los códigos de `5xx` estado de las respuestas de bloqueo AWS WAF personalizadas. Cuando una regla de ACL web devuelve una respuesta de bloqueo personalizada con un código de estado 500 o superior, Amazon Cognito responde con un código de `403 (Forbidden)` estado en su lugar. La respuesta sigue incluyendo los encabezados y el cuerpo de la respuesta personalizada. Este comportamiento se aplica de la siguiente manera:

- Páginas de inicio de sesión gestionadas: cuando el navegador carga una página de inicio de sesión gestionada, Amazon Cognito regresa `403` con los encabezados y el cuerpo de la respuesta personalizada. Las solicitudes que la página envía sin volver a cargarse, como el envío de formularios, reciben la respuesta en `403` bloque predeterminada sin los encabezados ni el cuerpo personalizados.
- Páginas de interfaz de usuario alojadas clásicas y otros puntos de enlace de grupos de usuarios: Amazon Cognito `403` regresa con los encabezados y el cuerpo de su respuesta personalizada para las páginas de interfaz de usuario alojadas clásicas. El proveedor de identidad y los puntos de enlace de la parte de confianza se comportan de la misma manera, independientemente de si su grupo de usuarios utiliza el inicio de sesión administrado o la marca clásica de la interfaz de usuario alojada.

- API de grupos de usuarios: la API de grupos de usuarios devuelve tu código de estado personalizado solo cuando está en el rango de 400 a 499, y devuelve el resto de códigos `403` de estado.

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 y Amazon Cognito AWS WAF . Para obtener información sobre AWS WAF los permisos, consulte los permisos de la AWS WAF API en la Guía AWS WAF para desarrolladores.

JSON

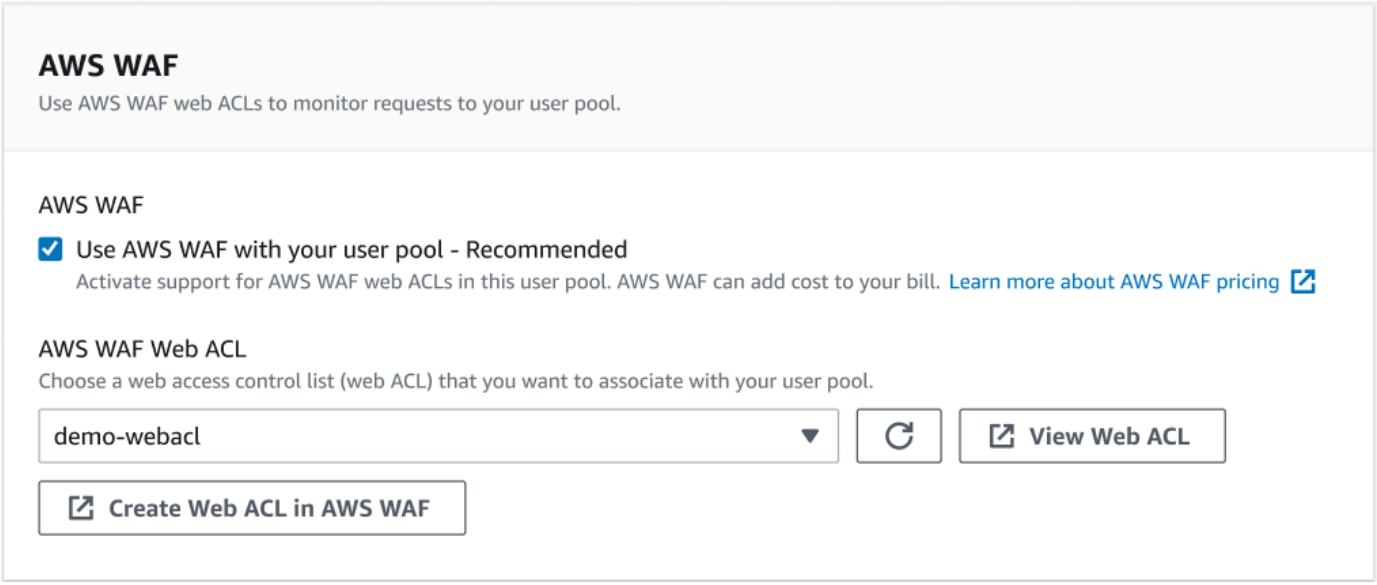
```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AllowWebACLUserPool",
   "Effect": "Allow",
   "Action": [
    "cognito-idp:ListResourcesForWebACL",
    "cognito-idp:GetWebACLForResource",
    "cognito-idp:AssociateWebACL"
   ],
   "Resource": [
    "arn:aws:cognito-idp:*:123456789012:userpool/*"
   ]
  },
  {
   "Sid": "AllowWebACLUserPoolWAFv2",
   "Effect": "Allow",
   "Action": [
    "wafv2:ListResourcesForWebACL",
    "wafv2:AssociateWebACL",
    "wafv2:DisassociateWebACL",
    "wafv2:GetWebACLForResource"
   ],
   "Resource": "arn:aws:wafv2:*:123456789012:*/webacl/*/*"
  },
  {
   "Sid": "DisassociateWebACL1",
```

```
    "Effect": "Allow",
    "Action": "wafv2:DisassociateWebACL",
    "Resource": "*"
   },
   {
    "Sid": "DisassociateWebACL2",
    "Effect": "Allow",
    "Action": [
     "cognito-idp:DisassociateWebACL"
    ],
    "Resource": [
     "arn:aws:cognito-idp:*:123456789012:userpool/*"
    ]
   }
  ]
}
```

Si bien debe conceder permisos de IAM, las acciones enumeradas son solo con permisos y no corresponden a una operación de la API.

Para activar 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. En la sección AWS WAF, haga clic en la sección Seguridad.
4. Elija Edit (Edición de).
5. 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 Consola de administración de AWS.
7. Seleccione Save changes (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.

Probando y registrando AWS WAF ACL web

Cuando estableces una acción de regla como Contar en tu 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 Pruebas y ajustes de sus protecciones de AWS WAF en la Guía para desarrolladores de AWS WAF .

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 inicio de sesión administrado 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 están disponibles en todos los planes de funciones de grupos de usuarios. Las características de seguridad de AWS WAF complementan la protección contra amenazas de Amazon Cognito. Puede activar las características en un grupo de usuarios. AWS WAF factura de forma individual 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 dirija sus 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 Consola de administración de AWS 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.

La falta de distinción entre mayúsculas y minúsculas no solo se aplica a las entradas de atributos, sino también a las salidas. Mixed-caseLos valores de los atributos en los grupos de usuarios que no distinguen entre mayúsculas y minúsculas se aplanan para convertirlos en minúsculas en la salida de texto del grupo de usuarios. Algunos ejemplos de salida de texto de grupos de usuarios son las respuestas de UserInfo, las respuestas a las consultas de los usuarios, como la salida de GetUser, y los eventos de entrada a los activadores de Lambda.

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 la distinción entre mayúsculas y minúsculas de los identificadores únicos y el SAML IdPs, consulte. Implementa el SP-initated inicio de sesión mediante SAML

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. Los grupos de usuarios que cree en la consola de Amazon Cognito no distinguen mayúsculas de minúsculas. Para crear un grupo de usuarios que distinga entre mayúsculas y minúsculas, debe utilizar la operación `CreateUserPool`. Antes del 12 de febrero de 2020, los grupos de usuarios distinguían entre mayúsculas y minúsculas de forma predeterminada, independientemente de la plataforma.

En el Sign-inmenú Consola de administración de AWS y en la `UsernameConfiguration` propiedad de DescribeUserPool, puede revisar la configuración de distinción entre mayúsculas y minúsculas de cada grupo de usuarios de su cuenta.

Migrar a un nuevo grupo de usuarios

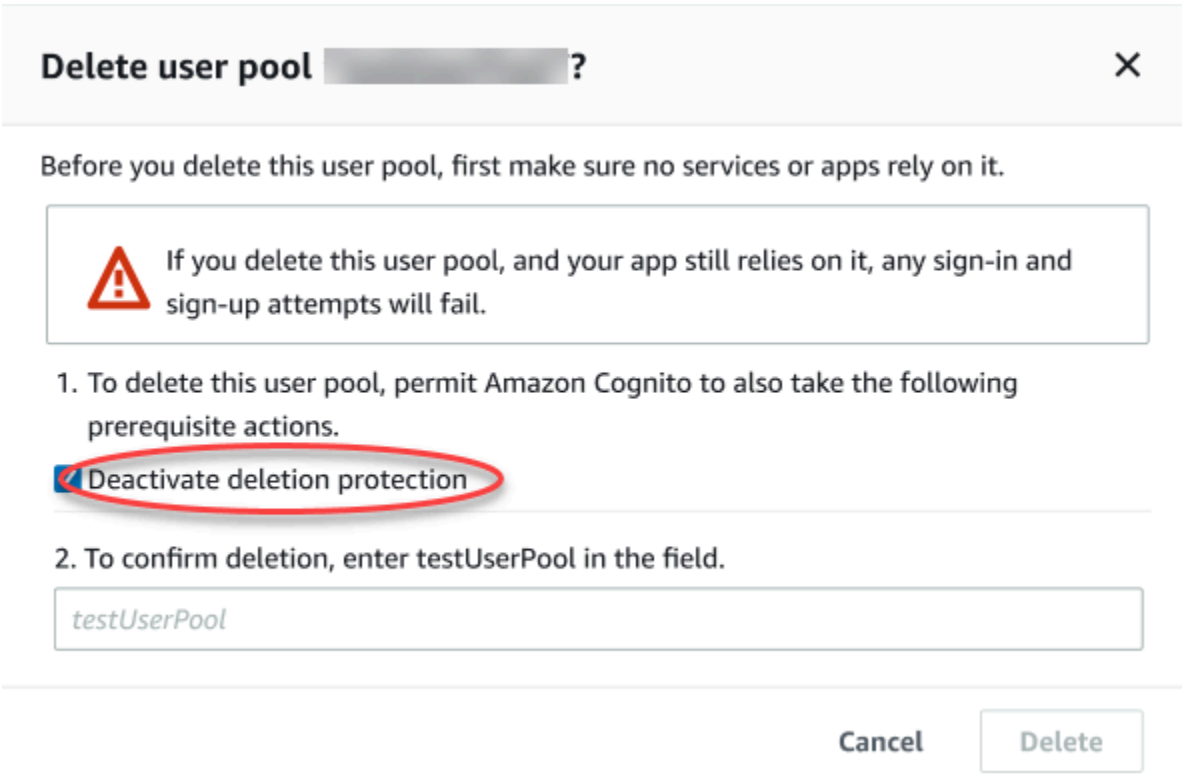
Debido a los posibles conflictos entre los perfiles de usuario, no se puede cambiar un grupo de usuarios de Amazon Cognito de una configuración que distingue entre mayúsculas y minúsculas a una que no hace la distinción. En su lugar, se deben migrar los usuarios a un nuevo grupo de usuarios. Debe crear 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 distingue entre mayúsculas y minúsculas, pero idénticos en un grupo de usuarios que no 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 contra eliminación activa, solo puede eliminar su grupo de usuarios después de confirmar que desea eliminarlo. Al eliminar un grupo de usuarios delConsola de administración de AWS, puede desactivar la protección contra la 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 de la API de Amazon Cognito, primero debe cambiarlo `DeletionProtection` a `Inactive` una UpdateUserPoolsolicitud. Si no desactiva la protección de eliminación, Amazon Cognito devuelve un error `InvalidParameterException`. Tras desactivar la protección contra la eliminación, puede eliminar el grupo de usuarios de una DeleteUserPoolsolicitud.

Amazon Cognito activa Deletion protection (Protección de eliminación) de forma predeterminada al crear un grupo de usuarios nuevo en la Consola de administración de AWS. 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 el SDK AWS CLI o con un AWS SDK, defina `True` el `DeletionProtection` parámetro en.

Puede activar o desactivar el estado de la protección de eliminación en el contenedor de Protección contra la eliminación del menú Configuración de la consola de Amazon Cognito.

Para configurar la protección de eliminación

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. Seleccione el menú Configuración, vaya a la pestaña de seguridad del grupo de usuarios y busque Deletion Protection. Seleccione Activar o 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 aplicación con la API de grupos de usuarios de Amazon Cognito, `PreventUserExistenceErrors` es `LEGACY` o se inhabilita de forma predeterminada. En la consola de Amazon Cognito, la opción Evitar errores de existencia de usuarios (un ajuste de `ENABLED` para `PreventUserExistenceErrors`) está seleccionada de forma predeterminada. Para actualizar la configuración de `PreventUserExistenceErrors`, realice una de las siguientes acciones:

- Cambie el valor de `PreventUserExistenceErrors` entre `ENABLED` y `LEGACY` en una solicitud de API UpdateUserPoolClient.
- Edite el cliente de la aplicación en la consola de Amazon Cognito y cambie el estado de Evitar errores de existencia de usuarios entre seleccionado (`ENABLED`) y deseleccionado (`LEGACY`).

Si el valor de esta propiedad es `LEGACY`, el cliente de la aplicación devuelve una respuesta de error `UserNotFoundException` cuando un usuario intenta iniciar sesión con un nombre de usuario que no existe en el grupo de usuarios.

Si el valor de esta propiedad es `ENABLED`, el cliente de la aplicación no revela con un error `UserNotFoundException` la inexistencia de una cuenta de usuario en el grupo de usuarios. Una configuración `PreventUserExistenceErrors` de `ENABLED` tiene los siguientes efectos cuando se envía una solicitud de un nombre de usuario que no existe:

- Amazon Cognito responde con información no específica a las solicitudes de la API cuando, si diera información específica, la respuesta podría revelar que existe un usuario válido.
- Amazon Cognito devuelve una respuesta genérica a un error de autenticación a las solicitudes de contraseña olvidada y a las solicitudes de autenticación con flujos de autenticación, excepto para

la autenticación basada en opciones (USER_AUTH), como USER_SRP_AUTH o CUSTOM_AUTH. La respuesta de error indica que el nombre de usuario o la contraseña son incorrectos.

- Amazon Cognito responde a las solicitudes de autenticación basada en opciones con una selección aleatoria de los tipos de desafíos permitidos para el grupo de usuarios. Es posible que su grupo de usuarios devuelva una clave de acceso, una contraseña de un solo uso o un desafío de contraseña.
- El comportamiento de las API de confirmación de cuenta y recuperación de contraseña de Amazon Cognito alterna entre devolver una respuesta que indica que se ha enviado un código a un medio de entrega simulado y devolver un error InvalidParameterException.

En la siguiente información se da información detallada sobre el comportamiento de las operaciones del grupo de usuarios cuando PreventUserExistenceErrors tiene la configuración ENABLED.

Operaciones de autenticación y creación de usuarios

Puede configurar las respuestas de error en la autenticación de nombre de usuario y contraseña y en la de contraseña remota segura (SRP). También puedes personalizar los errores que devuelvas con una autenticación personalizada. Choice-based la autenticación no se ve afectada por la PreventUserExistenceErrors configuración.

User-existence detalles de divulgación en los flujos de autenticación

Choice-based authentication

En el flujo de autenticación USER_AUTH basado en opciones, Amazon Cognito plantea un desafío a partir de los principales factores de autenticación disponibles, en función de la configuración del grupo de usuarios y de los atributos de los usuarios. Este flujo de autenticación puede devolver problemas relacionados con la contraseña, la contraseña remota segura (SRP), WebAuthn la clave de paso, la contraseña de un solo uso (OTP) por SMS o la OTP de correo electrónico. Con PreventUserExistenceErrors activo, Amazon Cognito envía un desafío a los usuarios inexistentes para que completen una o más de las formas de autenticación disponibles. Si PreventUserExistenceErrors está inactivo, Amazon Cognito devuelve una excepción UserNotFound.

Autenticación de nombre de usuario y contraseña

Los flujos de autenticación ADMIN_USER_PASSWORD_AUTH y USER_PASSWORD_AUTH y el flujo PASSWORD de USER_AUTH devuelven una NotAuthorizedException con el mensaje

`Incorrect username or password` cuando `PreventUserExistenceErrors` está activo. Cuando `PreventUserExistenceErrors` está inactivo, estos flujos devuelven `UserNotFoundException`.

Autenticación basada en contraseña remota segura (SRP)

Como práctica recomendada, implemente solo `PreventUserExistenceErrors` con `USER_SRP_AUTH` o el flujo `PASSWORD_SRP` de `USER_AUTH` en grupos de usuarios sin los atributos de alias de dirección de correo electrónico, número de teléfono o nombre de usuario preferido. Es posible que los usuarios con atributos de alias no estén sujetos a la supresión de la existencia del usuario en el flujo de autenticación del SRP. Username-password los flujos de autenticación (`ADMIN_USER_PASSWORD_AUTH`y el `USER_AUTH` `PASSWORD` desafío) suprimen por completo la existencia de usuarios a partir de los atributos de alias. `USER_PASSWORD_AUTH`

Cuando alguien intenta iniciar sesión con SRP con un nombre de usuario que el cliente de aplicación no conoce, Amazon Cognito devuelve una respuesta simulada en el primer paso, tal y como se describe en RFC 5054. Amazon Cognito devuelve el mismo fragmento salt y un ID de usuario interno en formato de 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. Para obtener más información sobre cómo implementar una autenticación personalizada, consulte Sign-in con contraseñas persistentes y una carga segura.

> **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 del desafío de autenticación personalizada

Amazon Cognito invoca los desencadenadores de Lambda de desafío de autenticación personalizada cuando los usuarios intentan iniciar sesión con el flujo de autenticación `CUSTOM_AUTH`, pero no se encuentra su nombre de usuario. El evento de entrada incluye un parámetro booleano denominado `UserNotFound` con un valor de `true` para cualquier usuario inexistente. Este parámetro aparece en los eventos de solicitud que el grupo de usuarios envía a las funciones de Lambda de creación, definición y verificación de autenticación que componen la

arquitectura de autenticación personalizada. Al examinar este indicador en la lógica de la función de Lambda, puede simular desafíos de autenticación personalizados para un usuario que no exista.

Desencadenador de Lambda Antes de la autenticación

Amazon Cognito invoca el desencadenador antes de la autenticación cuando los usuarios intentan iniciar sesión, pero no se encuentra su nombre de usuario. El evento de entrada incluye un parámetro `UserNotFound` con un valor de `true` para cualquier usuario inexistente.

En la siguiente lista se describe el efecto de `PreventUserExistenceErrors` en la creación de cuentas de usuario.

User-existence detalles de divulgación en los flujos de creación de usuarios

SignUp

La operación `SignUp` siempre devuelve `UsernameExistsException` cuando ya existe 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 Obligación de 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 Importación de usuarios con un desencadenador de Lambda para la migración de usuarios.

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`.

Creación de aplicaciones B2B multiusuario

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. La multitenencia aporta varias ventajas más, como 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.

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 escalar verticalmente esta capacidad cuando la base de clientes crezca, puede adquirir capacidad de cuota adicional.

> **Note**
>
> Las cuotas de Amazon Cognito se aplican por Cuenta de AWS y. Región de AWS Todos los inquilinos de la aplicación las comparten. Revise las cuotas de servicio de Amazon Cognito y asegúrese de que puedan dar respuesta al volumen y el número de inquilinos previstos para su aplicación.

Single-Region La multitenencia también puede suponer ventajas para sus clientes y administradores.

En la siguiente lista se describen algunas de las ventajas de la multitenencia con recursos compartidos.

Ventajas de la multitenencia

Directorio común de usuarios

Multi-tenancy 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 sean exclusivos del inquilino, cualquier estrategia de multitenencia con un único grupo de usuarios tendrá 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 protección contra amenazas, autenticación multifactor (MFA) y estándares de AWS WAF 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. Cuando quiera mantener una configuración de seguridad homogénea en las aplicaciones de Amazon Cognito de varias regiones, aplique 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 desencadenadores 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 de Lambda compartidas pueden aplicar estándares para flujos de autenticación personalizados, la migración de usuarios, la generación de tokens y otras funciones dentro de una misma región.

Mensajes comunes

Para poder enviar mensajes SMS a los usuarios, es preciso añadir la configuración de Amazon Simple Notification Service (Amazon SNS) en la región. Puede enviar mensajes de correo electrónico con identidades y dominios verificados de Amazon Simple Email Service (Amazon SES) incluidos en una región.

Con la multitenencia, puede compartir esta sobrecarga de configuración y mantenimiento entre todos los inquilinos. Dado que Amazon SNS y Amazon SES no están disponibles en todas las Regiones de AWS, es necesario prestar especial atención cuando se plantee la posibilidad de dividir los recursos entre regiones.

Cuando utiliza proveedores de mensajes personalizados, obtiene la personalización común de una sola función de Lambda para administrar la entrega de mensajes.

El inicio de sesión administrado establece una cookie de sesión en el navegador para que reconozca a un usuario que ya se ha autenticado. Cuando autentica a usuarios locales en un grupo de usuarios, la cookie de sesión los autentica para todos los clientes de aplicación 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 los clientes de aplicación con una cookie de sesión de interfaz de usuario alojada.

- Puede distribuir a los usuarios en grupos de usuarios por inquilino.
- Puede sustituir el inicio de sesión de la interfaz de usuario alojada por el inicio de sesión de la API de grupos de usuarios de Amazon Cognito.

Temas

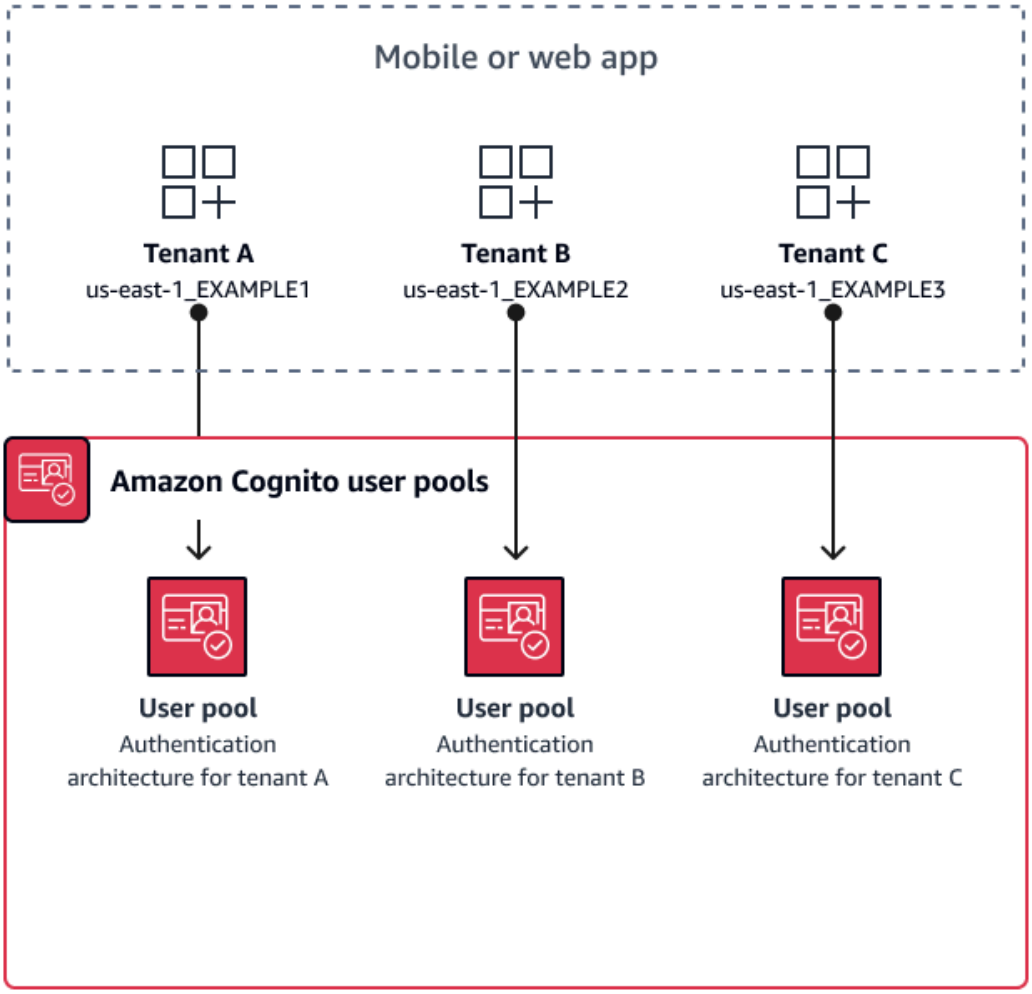
- User-pool prácticas recomendadas para la multitenencia
- App-client mejores prácticas de tenencia múltiple
- Prácticas recomendadas para la multitenencia de grupos de usuarios
- Custom-attribute prácticas recomendadas para la multitenencia
- Prácticas recomendadas de multitenencia con ámbito personalizado
- Multi-tenancy recomendaciones de seguridad

User-pool prácticas recomendadas para la multitenencia

Cree un grupo de usuarios para cada inquilino de la aplicación. Este enfoque aporta el máximo aislamiento a cada inquilino. Puede implementar diferentes configuraciones para cada inquilino. El aislamiento de los inquilinos por grupos de usuarios le ofrece la flexibilidad de asignar usuarios a inquilinos. 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.

Utilice este enfoque para configurar la interfaz de usuario alojada de cada inquilino por separado y redirigir a los usuarios a la instancia específica del inquilino que les corresponda en su aplicación. También puedes usar este enfoque para integrarte con servicios de back-end como Amazon API Gateway.

En el siguiente diagrama, se muestra cada inquilino con un grupo de usuarios dedicado.



Cuándo implementar la multitenencia de grupos de usuarios

Cuando el aislamiento y la personalización sean sus principales intereses. La relación entre los usuarios y los inquilinos puede ser compleja en una arquitectura con varios grupos de usuarios. Supongamos, por ejemplo, que tiene dos inquilinos del ámbito de la educación. 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 necesite una MFA en una aplicación, pero no en otra, o necesite tener una política de contraseñas diferente. Dado que los usuarios locales pueden iniciar sesión en varios clientes de aplicación en grupos de usuarios con inicio de sesión administrado, la multitenencia de grupos de usuarios también es ideal cuando queremos que más de uno de los inquilinos inicie sesión con el inicio de sesión administrado.

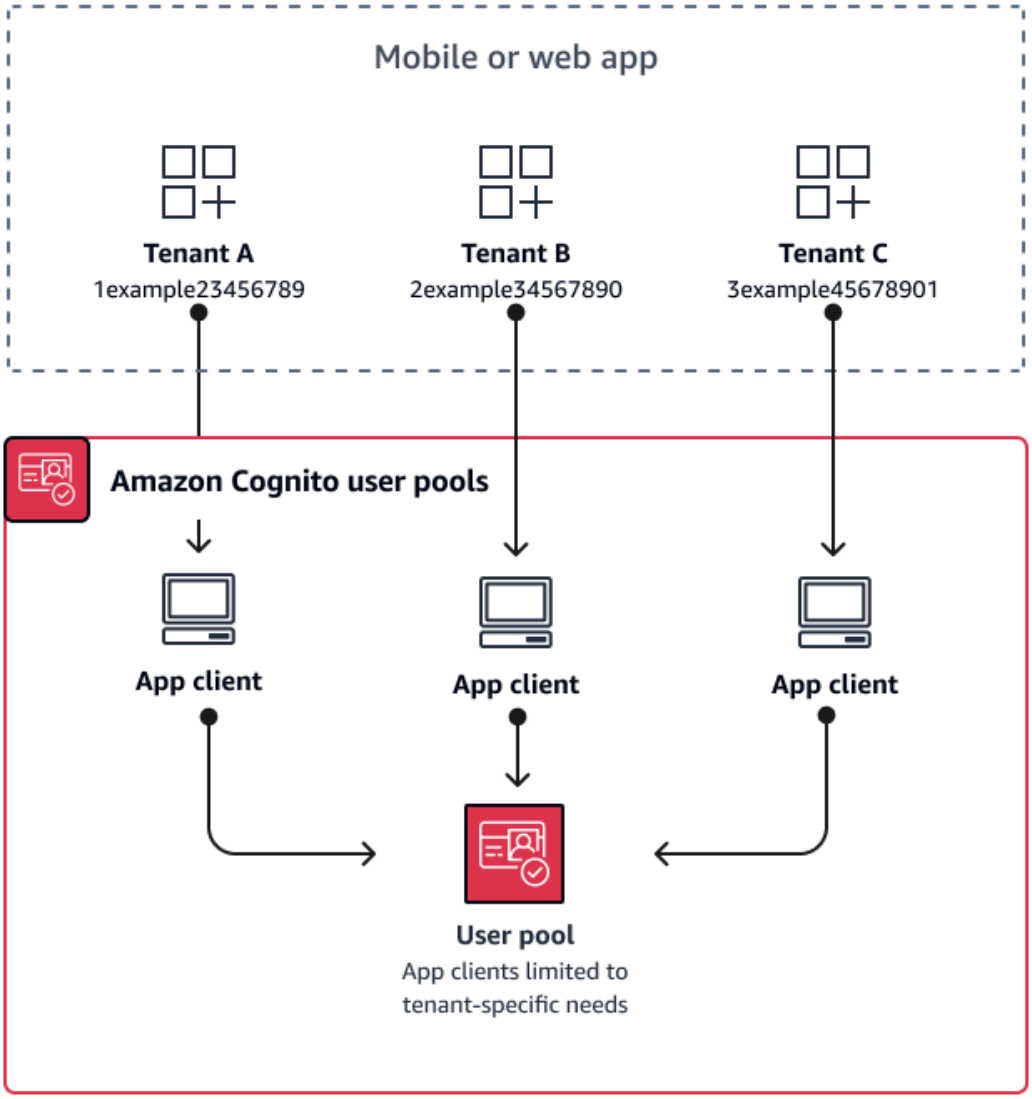
Nivel de esfuerzo

El nivel de esfuerzo de desarrollo y operación en este enfoque es alto. Para garantizar resultados coherentes y predecibles para la familia de aplicaciones a medida que la arquitectura de autenticación se vuelve más compleja, integre los recursos de Amazon Cognito con las herramientas de automatización y mantenga las bases de referencia. Cuando quiera crear un punto de partida único para sus aplicaciones, deberá crear los elementos de la interfaz de usuario (IU) para captar la decisión inicial que dirige a los usuarios al recurso correcto.

App-client mejores prácticas de tenencia múltiple

Cree un cliente de aplicación por cada inquilino de su aplicación. Con la multitenencia de clientes de aplicación, puede asignar cualquier usuario a clientes de aplicación vinculados a inquilinos y retener al mismo tiempo un perfil de usuario único. 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 cada inquilino con un cliente de aplicación dedicado en un grupo de usuarios compartido.



Cuándo implementar la multitenencia de clientes de aplicación

Cuando pueda elegir una configuración universal para los ajustes del grupo de usuarios, como los desencadenadores 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 aplicación, la multitenencia de app-cliente es ideal para iniciar sesión con aplicaciones específicas de cada cliente IdPs o con la API de grupos de usuarios de Amazon Cognito. App-client La multitenencia también es adecuada para entornos de uno a varios en los que se quiere permitir a los usuarios realizar la transición entre varias aplicaciones.

Nivel de esfuerzo

App-client la multitenencia requiere un esfuerzo moderado. Uno de los principales retos que plantea la multitenencia de clientes de aplicación 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 multitenencia de cliente de aplicación, evite el inicio de sesión en una interfaz de usuario alojada cuando sea necesario mantener el aislamiento. Puede distribuir su aplicación móvil o los enlaces a su aplicación web con la lógica del cliente de aplicación integrada, o puede 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.

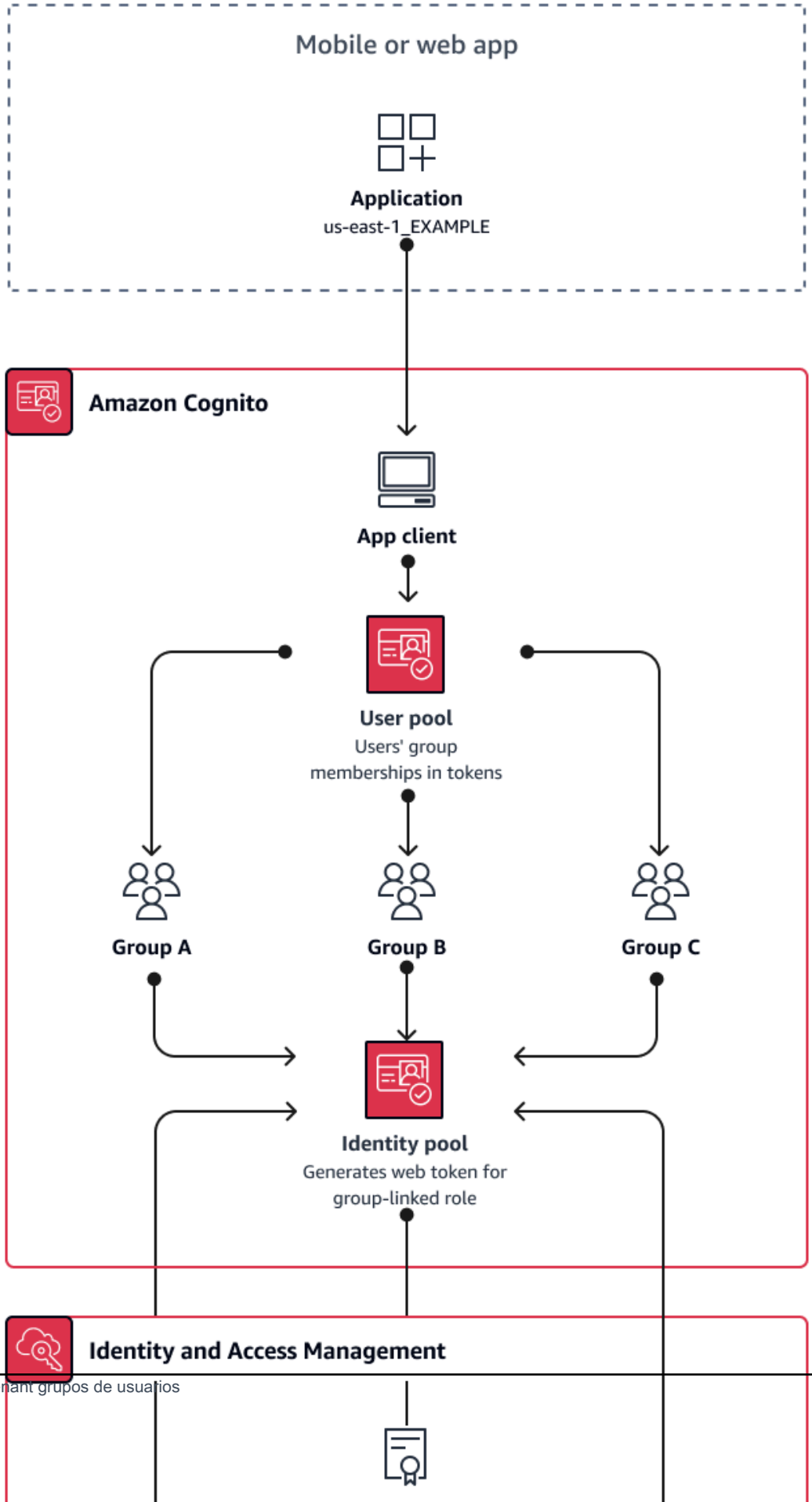
Prácticas recomendadas para la multitenencia de grupos de usuarios

Group-based La multitenencia funciona mejor cuando la arquitectura requiere grupos de usuarios de Amazon Cognito con grupos de identidades.

Los tokens de ID y acceso del grupo de usuarios contienen una notificación `cognito:groups`. Además, los tokens de ID contienen notificaciones `cognito:roles` y `cognito:preferred_role`. Cuando el resultado principal de la autenticación en su aplicación sea la obtención de credenciales temporales de AWS de un grupo de identidades, la pertenencia a un grupo de usuarios puede determinar el rol de IAM y los permisos que reciban.

Como ejemplo, pensemos en tres inquilinos, cada uno de los cuales almacena recursos 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 muestran inquilinos que comparten un cliente de aplicación y un grupo de usuarios, con grupos específicos del 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 group-linked role
Identity and Access Management

Cuándo implementar la multitenencia de grupo

Cuando el acceso a AWS los recursos es su principal preocupación. Los grupos incluidos en los grupos de usuarios de Amazon Cognito son un mecanismo de control de acceso basado en roles (RBAC). Puede configurar numerosos 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 otras notificaciones en tokens de un usuario.

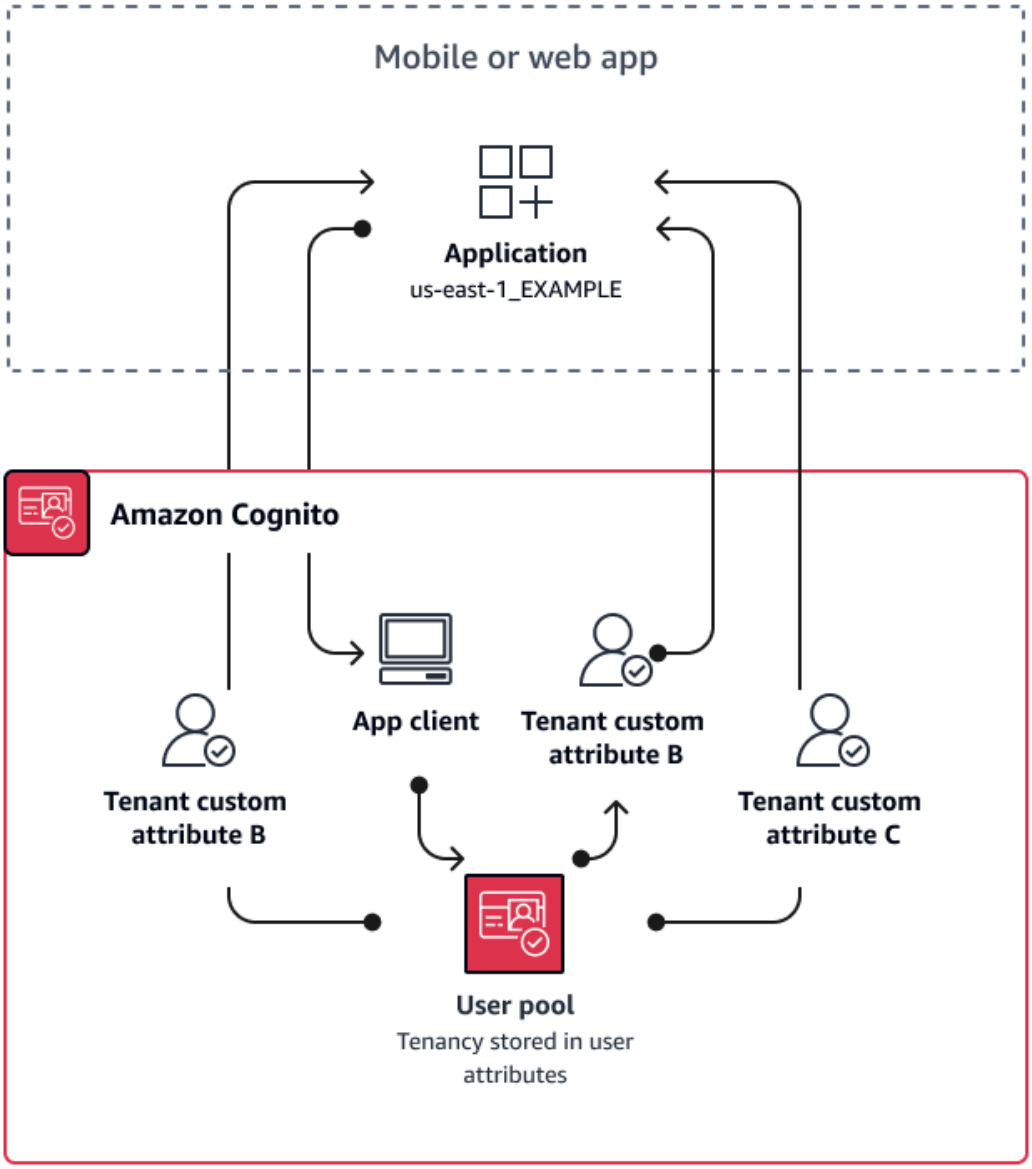
Nivel de esfuerzo

El nivel de esfuerzo para mantener la multitenencia solo con la pertenencia a un grupo es bajo. Sin embargo, para ampliar el rol de los grupos incluidos en 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. Puede integrar Amazon Verified Permissions con las aplicaciones para tomar decisiones de autorización 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.

Custom-attribute prácticas recomendadas para la multitenencia

Amazon Cognito admite atributos personalizados con los nombres que elija. Los atributos personalizados son útiles cuando, por ejemplo, distinguen la tenencia de los usuarios en un grupo de usuarios compartido. Cuando asigna a los usuarios un valor para un atributo como `custom:tenantID`, la aplicación puede asignar el acceso a los recursos específicos del inquilino en función de dicho atributo. Un atributo personalizado que defina un ID de inquilino debe ser inmutable o de solo lectura para el cliente de aplicación.

En el siguiente diagrama se muestran los inquilinos que comparten un cliente de aplicación 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, puede distribuir una aplicación o URL de inicio de sesión únicos. Una vez que el usuario inicie sesión, la aplicación podrá procesar la `custom:tenantID` reclamación para determinar qué activos cargar, qué marca aplicar y qué funciones mostrar. Para tomar decisiones avanzadas de control de acceso a partir de los atributos de usuario, configure el 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 la tenencia esté en superficie. Un atributo de inquilino puede contribuir a los resultados de la marca y el diseño. Si quiere lograr un buen aislamiento entre los inquilinos, los atributos personalizados no son la mejor opción. Cualquier diferencia entre los inquilinos que deba configurarse en el grupo de usuarios o en el ámbito de aplicación y cliente, como la MFA o la marca de la interfaz de usuario alojada, requiere la creación de distinciones entre los inquilinos de un modo que los atributos personalizados no ofrezcan. Con los grupos de identidades, puede incluso elegir el rol de IAM entre los usuarios a partir de la notificación del atributo personalizado que se incluye en su token de ID.

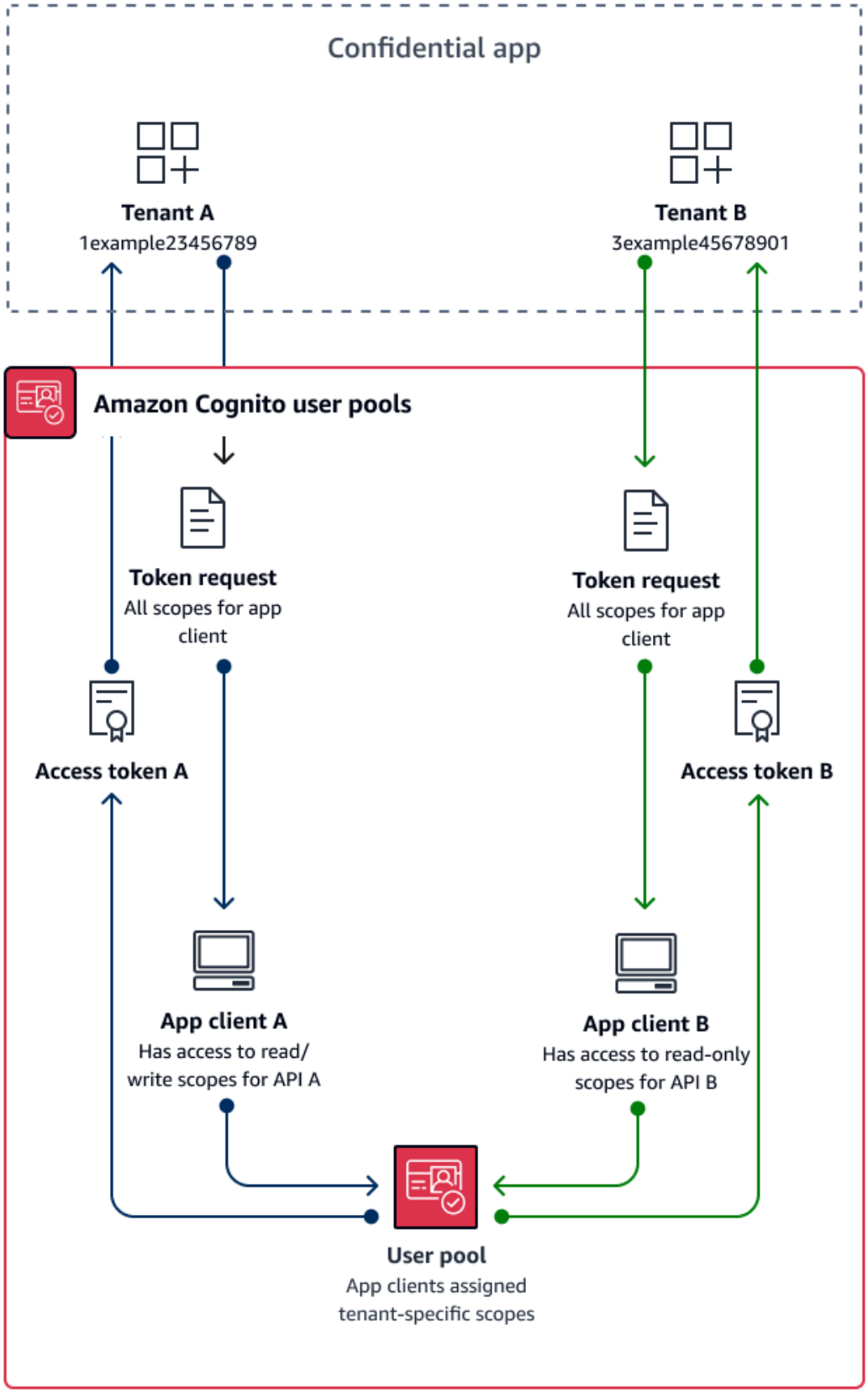
Nivel de esfuerzo

Dado que la multitenencia con atributos personalizados transfiere la responsabilidad de las decisiones de autorización basadas en el inquilino a la aplicación, el nivel de esfuerzo suele ser elevado. Si ya conoce bien una configuración de cliente que analiza las notificaciones de OIDC o, en Amazon Verified Permissions, es posible que este enfoque requiera el menor esfuerzo.

Prácticas recomendadas de multitenencia con ámbito personalizado

Amazon Cognito admite ámbitos de OAuth 2.0 personalizados para servidores de recursos. Puede implementar la multitenencia de clientes de aplicaciones en grupos de usuarios para modelos de autorización de máquina a máquina (M2M) con ámbitos personalizados. Scope-based La multitenencia reduce el esfuerzo necesario para implementar la multitenencia M2M al definir el acceso en el cliente de la aplicación o en la configuración de la aplicación.

En el siguiente diagrama, se muestra una opción de multitenencia con ámbito personalizado. Muestra cada inquilino con un cliente de aplicación dedicado que tiene acceso a los ámbitos relevantes de un grupo de usuarios.


Confidential app
Tenant A
1example23456789
Tenant B
3example45678901
Amazon Cognito user pools
Token request
All scopes for app client
Token request
All scopes for app client
Access token A
Access token B
App client A
Has access to read/ write scopes for API A
App client B
Has access to read-only scopes for API B
User pool
App clients assigned tenant-specific scopes

Cuándo implementar la multitenencia con ámbitos personalizados

Cuando se utiliza una autorización de M2M con credenciales de cliente en un cliente confidencial. Como práctica recomendada, cree servidores de recursos que sean exclusivos para un cliente de aplicación. La multitenencia con ámbitos personalizados puede depender de la solicitud o del cliente.

Request-dependent

Implemente la lógica de la aplicación para solicitar solo los ámbitos que coincidan con los requisitos del inquilino. Por ejemplo, un cliente de aplicación podría conceder acceso de lectura y escritura a las API A y B, pero la aplicación A del inquilino solo solicita el ámbito de lectura de la API A y el ámbito que indica la tenencia. Este modelo permite combinaciones más complejas de ámbitos compartidos entre inquilinos.

Client-dependent

Solicite todos los ámbitos asignados a un cliente de aplicación en las solicitudes de autorización. Para ello, omita el parámetro de solicitud `scope` en la solicitud al Punto de conexión de token. Si obtiene tokens de acceso M2M con la operación de GetClientTokenAPI, omita el parámetro para obtener el mismo resultado. `Scopes` Este modelo permite a los clientes de aplicación almacenar los indicadores de acceso que desee añadir a los ámbitos personalizados.

En cualquier caso, las aplicaciones reciben tokens de acceso con ámbitos que indican sus privilegios en relación con los orígenes de datos de los que dependen. Los ámbitos también pueden presentar otro tipo de información a la aplicación, como:

- Designar tenencias
- Contribuir al registro de solicitudes
- Indicar las API que la aplicación está autorizada a consultar
- Informar de las comprobaciones iniciales a los clientes activos

Nivel de esfuerzo

Custom-scope La multitenencia requiere un nivel de esfuerzo variable en relación con la escala de la aplicación. Debe diseñar una lógica de aplicación que permita a las aplicaciones analizar los tokens de acceso y presentar las solicitudes de API adecuadas.

Por ejemplo, el ámbito de un servidor de recursos viene con el formato `[resource server identifier]/[name]`. Es poco probable que el identificador del servidor de recursos sea

relevante para la decisión de autorización del ámbito del inquilino, por lo que es necesario analizar el nombre del ámbito de forma coherente.

Ejemplo de recurso

La siguiente AWS CloudFormation plantilla crea un grupo de usuarios para la multitenencia de ámbito personalizado con un servidor de recursos y un cliente de aplicaciones.

```
AWSTemplateFormatVersion: "2010-09-09"
Description: A sample template illustrating scope-based multi-tenancy
Resources:
  MyUserPool:
    Type: "AWS::Cognito::UserPool"
  MyUserPoolDomain:
    Type: AWS::Cognito::UserPoolDomain
    Properties:
      UserPoolId: !Ref MyUserPool
      # Note that the value for "Domain" must be unique across all of AWS.
      # In production, you may want to consider using a custom domain.
      # See: https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-
pools-add-custom-domain.html#cognito-user-pools-add-custom-domain-adding
      Domain: !Sub "example-userpool-domain-${AWS::AccountId}"
  MyUserPoolResourceServer:
    Type: "AWS::Cognito::UserPoolResourceServer"
    Properties:
      Identifier: resource1
      Name: resource1
      Scopes:
        - ScopeDescription: Read-only access
          ScopeName: readScope
      UserPoolId: !Ref MyUserPool
  MyUserPoolTenantBatch1ResourceServer:
    Type: "AWS::Cognito::UserPoolResourceServer"
    Properties:
      Identifier: TenantBatch1
      Name: TenantBatch1
      Scopes:
        - ScopeDescription: tenant1 identifier
          ScopeName: tenant1
        - ScopeDescription: tenant2 identifier
          ScopeName: tenant2
      UserPoolId: !Ref MyUserPool
  MyUserPoolClientTenant1:
```

```
    Type: "AWS::Cognito::UserPoolClient"
    Properties:
      AllowedOAuthFlows:
        - client_credentials
      AllowedOAuthFlowsUserPoolClient: true
      AllowedOAuthScopes:
        - !Sub "${MyUserPoolTenantBatch1ResourceServer}/tenant1"
        - !Sub "${MyUserPoolResourceServer}/readScope"
      GenerateSecret: true
      UserPoolId: !Ref MyUserPool
Outputs:
  UserPoolClientId:
    Description: User pool client ID
    Value: !Ref MyUserPoolClientTenant1
  UserPoolDomain:
    Description: User pool domain
    Value: !Sub "https://${MyUserPoolDomain}.auth.${AWS::Region}.amazoncognito.com"
```

Multi-tenancy recomendaciones de seguridad

Para garantizar que su aplicación sea más segura le recomendamos lo siguiente:

- Valide la tenencia en la aplicación con Amazon Verified Permissions. Cree políticas que examinen los derechos del grupo de usuarios, los clientes de aplicaciones, 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 la multitenencia.
- 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 inmutables o de solo lectura para los atributos personalizados de perfil de usuario que identifiquen a los inquilinos. Solo puede establecer el valor de los atributos inmutables al crear un usuario o cuando un usuario se registre en su grupo de usuarios. Además, proporcione a los clientes de aplicaciones acceso de solo lectura a los atributos.
- Aplique una asignación de uno a uno entre el IdP externo de un inquilino y el cliente de 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.

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

La implementación de Amazon Cognito es una combinación de Consola de administración de AWS herramientas administrativas del AWS SDK y bibliotecas del SDK en las aplicaciones. 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 integración más sencilla que puede crear con los grupos de usuarios de Amazon Cognito es el inicio de sesión administrado. El inicio de sesión administrado es una aplicación de inicio de sesión basada en la web lista para usar que permite probar e implementar rápidamente los grupos de usuarios de Amazon Cognito. La autenticación de grupos de usuarios con inicio de sesión administrado requiere bibliotecas OpenID Connect (OIDC) que dirigen a los usuarios a las páginas de inicio de sesión alojadas. En esta serie de puntos de conexión web redireccionados e interactivos para el usuario, Amazon Cognito gestiona el flujo de autenticación, lo que incluye el inicio de sesión de terceros, la autenticación multifactor (MFA) y la elección de un flujo de autenticación. Su aplicación solo tiene que procesar el resultado de autenticación que Amazon Cognito devuelve en la respuesta.

También puede añadir un AWS SDK a su aplicación, crear interfaces de autenticación personalizadas e invocar operaciones de API para autenticar y autorizar a sus usuarios. AWS Amplifyes Servicio de AWS para crear aplicaciones completas, con la autenticación de Amazon Cognito en el back-end.

Por ejemplo, es posible que la aplicación invoque el inicio de sesión administrado 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. Los grupos de identidades no tienen las mismas opciones de autenticación administrada que los grupos de usuarios: para acceder a las AWS credenciales de sus aplicaciones, implemente las operaciones de los grupos de identidades en los módulos del SDK importados. 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 para los AWS SDK, consulte. Autenticación con AWS SDK 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 las operaciones de la API de Amazon Cognito y su sintaxis.

> **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 Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

Temas

- Autenticación con AWS Amplify
- Autenticación con AWS SDK
- Funcionamiento de la autenticación con Amazon Cognito
- Uso de este servicio con un AWS SDK
- Autorización con Amazon Verified Permissions

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.

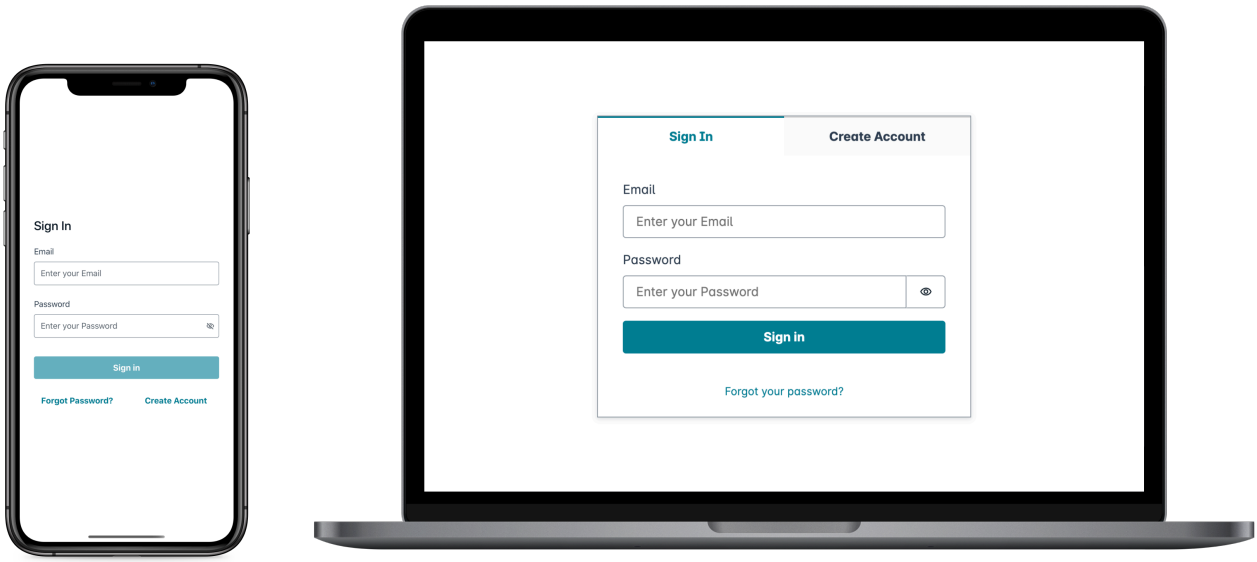
- Autenticación con Amplify para React
- Autenticación con Amplify para React Native
- Autenticación con Amplify para Swift (iOS)
- Autenticación Amplify para Android
- Amplificar la autenticación para Flutter

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

Inicio de sesión administrado de grupos de usuarios puede satisfacer las necesidades esenciales de un frontend de autenticación para una aplicación web o móvil. A fin de personalizar su interfaz de usuario (UI) más allá de los parámetros que admite el inicio de sesión administrado, 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 AWS SDK

Si desea utilizar un back-end 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

Funcionamiento de la autenticación con Amazon Cognito

Cuando un cliente inicia sesión en un grupo de usuarios de Amazon Cognito, su aplicación recibe 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 AWS credenciales temporales.

Con el inicio de 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 (IU), puede invocar una interfaz de usuario web prediseñada (el inicio de sesión administrado) o la página de inicio de sesión de su proveedor de identidades (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 de grupos de usuarios con inicio de sesión administrado
- Autenticación y autorización de la API de grupo de usuarios con un AWS SDK
- Autenticación del grupo de usuarios con un proveedor de identidades de terceros
- Autenticación del grupo de identidades

Autenticación de grupos de usuarios con inicio de sesión administrado

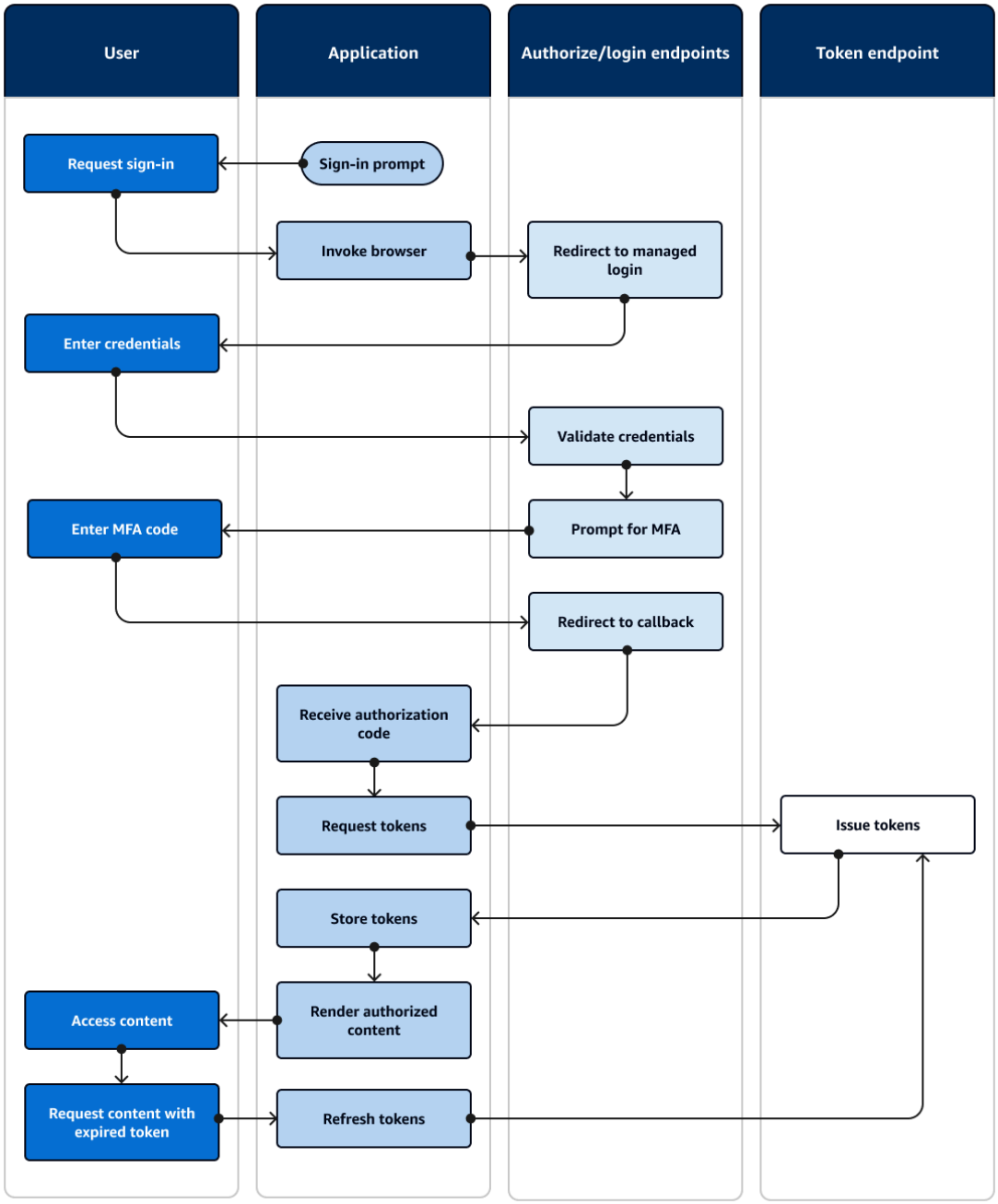
El inicio de sesión administrado es un sitio web que está vinculado a su grupo de usuarios y al cliente de aplicación. Puede realizar operaciones de inicio de sesión, registro y restablecimiento de contraseñas para los usuarios. La implementación de una aplicación con un componente de inicio de sesión administrado 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 en el navegador del usuario las páginas web del inicio de sesión administrado.

Las aplicaciones recopilan los JWT de los usuarios con una ubicación de redireccionamiento web o de aplicación. Las aplicaciones que implementan el inicio de sesión administrado pueden conectarse a grupos de usuarios para tareas de autenticación como si fueran un IdP OpenID Connect (OIDC).

El inicio de sesión administrado encaja bien en un modelo en el que las aplicaciones requieren los servicios de autenticación de un servidor de autorización OIDC, pero no necesitan inmediatamente características 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.

El inicio de sesión administrado 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 ámbitos de OAuth 2.0.

En el siguiente diagrama, se ilustra un inicio de sesión típico para una autenticación de la API.



Flujo de autenticación del inicio de sesión administrado

1. El usuario accede a su aplicación.
2. Selecciona un enlace para iniciar sesión.
3. La aplicación dirige al usuario a una petición de inicio de sesión en las páginas de inicio de sesión administrado del dominio de su grupo de usuarios.
4. Introduce su nombre de usuario y contraseña.

5. El grupo de usuarios valida las credenciales del usuario y determina si el usuario ha activado la autenticación multifactor (MFA).
6. La página del inicio de sesión administrado solicita al usuario que introduzca un código de MFA.
7. El usuario introduce el código de la MFA.
8. El grupo de usuarios redirige al usuario a la URL de la aplicación.
9. La aplicación toma el código de autorización del parámetro de solicitud de URL que el inicio de sesión administrado ha añadido a la URL de devolución de llamada.
10. La aplicación solicita tokens con el código de autorización.
11. El punto de conexión del token devuelve los JWT a la aplicación.
12. La aplicación descodifica, valida y guarda 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 caduca 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 de conexión del token con el token de actualización.

Variantes y personalización

Puede personalizar el aspecto de sus páginas de inicio de sesión administrado con el editor de marcas para todo su grupo de usuarios o en cualquier cliente de aplicación. También puede configurar los clientes de aplicación con sus propios proveedores de identidades, ámbitos, acceso a los atributos de usuario y configuración de seguridad avanzada.

Recursos relacionados

- Inicio de sesión administrado de grupos de usuarios
- Ámbitos, M2M y servidores de recursos
- Referencia de los puntos de conexión de grupos de usuarios y del inicio de sesión administrado

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

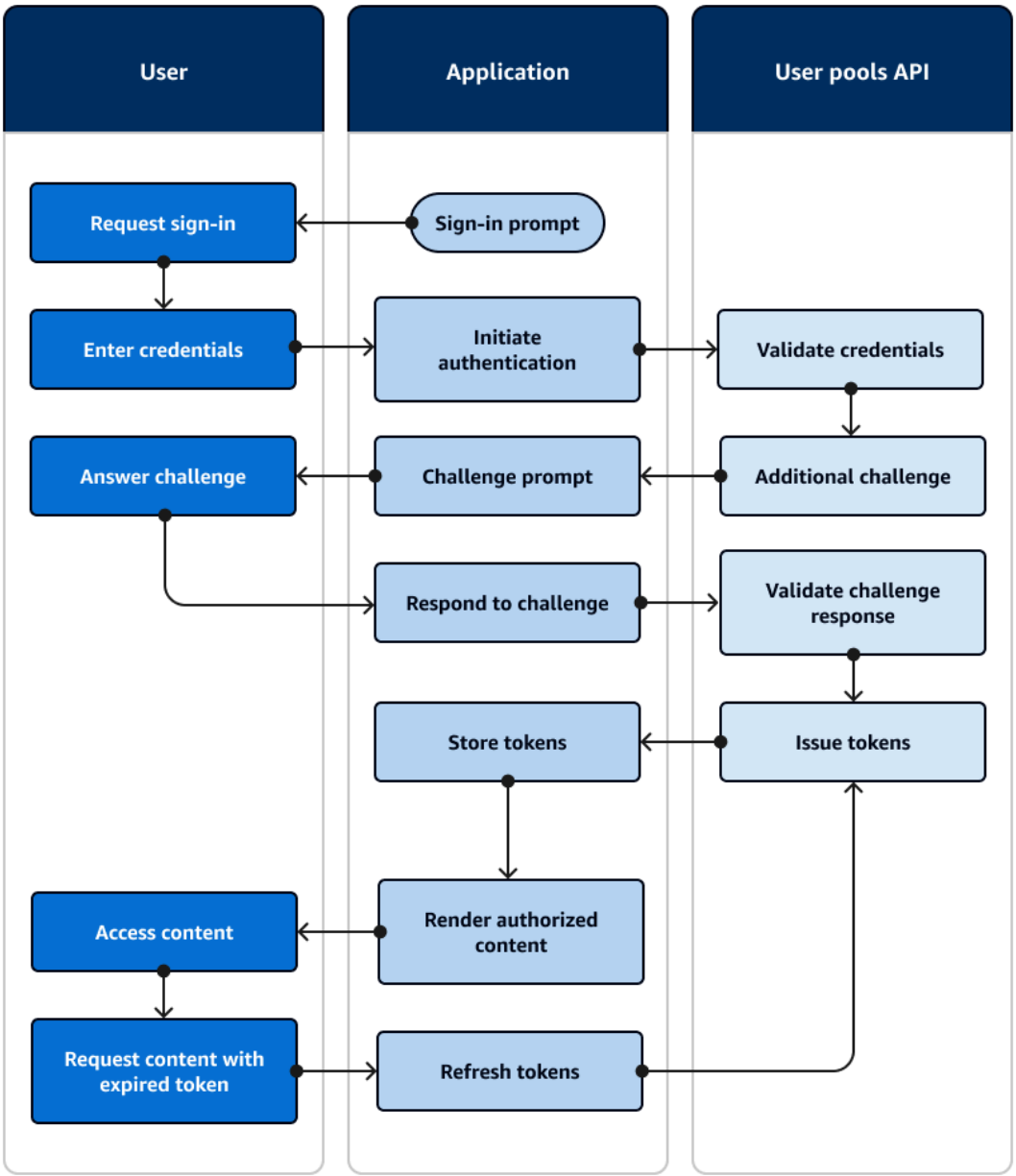
AWS ha desarrollado componentes para los grupos de usuarios de Amazon Cognito, o el proveedor de identidad de Amazon Cognito, en diversos marcos de desarrollo. Los métodos integrados en estos SDK llaman a la API de grupos de usuarios de Amazon Cognito. El mismo espacio de nombres de la API de grupos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

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

Esta aplicación no necesita desplegar una implementación de relación de confianza OpenID Connect (OIDC) completa. En su lugar, tiene la capacidad de descodificar y utilizar los JWT. Cuando desee que los usuarios locales accedan al conjunto completo de características del grupo de usuarios, cree la 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 de una autenticación de API, modifique el token en tiempo de ejecución con un Desencadenador de Lambda anterior a la generación del token.

En el siguiente diagrama, se ilustra un inicio de sesión típico para una autenticación de la API.



Flujo de autenticación de la API

1. El usuario accede a su aplicación.
2. Selecciona un enlace para iniciar sesión.
3. Introduce su nombre de usuario y contraseña.
4. La aplicación invoca el método que realiza una InitiateAuthsolicitud a la API. La solicitud pasa las credenciales del usuario a un grupo de usuarios.

5. El grupo de usuarios valida las credenciales del usuario y determina si 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 una petición que recopila el código MFA del usuario.
8. La aplicación invoca el método que realiza una solicitud a la RespondToAuthChallengeAPI. La solicitud pasa el código de la MFA del usuario.
9. El grupo de usuarios valida el código de la MFA del usuario.
10. El grupo de usuarios responde con los JWT del usuario.
11. La aplicación descodifica, valida y guarda 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 caduca 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 incrementar este flujo con desafíos adicionales, por ejemplo, desafíos de autenticación personalizados propios. 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 (de cliente) y confidenciales (de servidor) duplicadas.

Recursos relacionados

- API de grupos de usuarios de Amazon Cognito
- Creación de una nueva aplicación en la consola de Amazon Cognito
- Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles
- Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado

Autenticación del grupo de usuarios con un proveedor de identidades de terceros

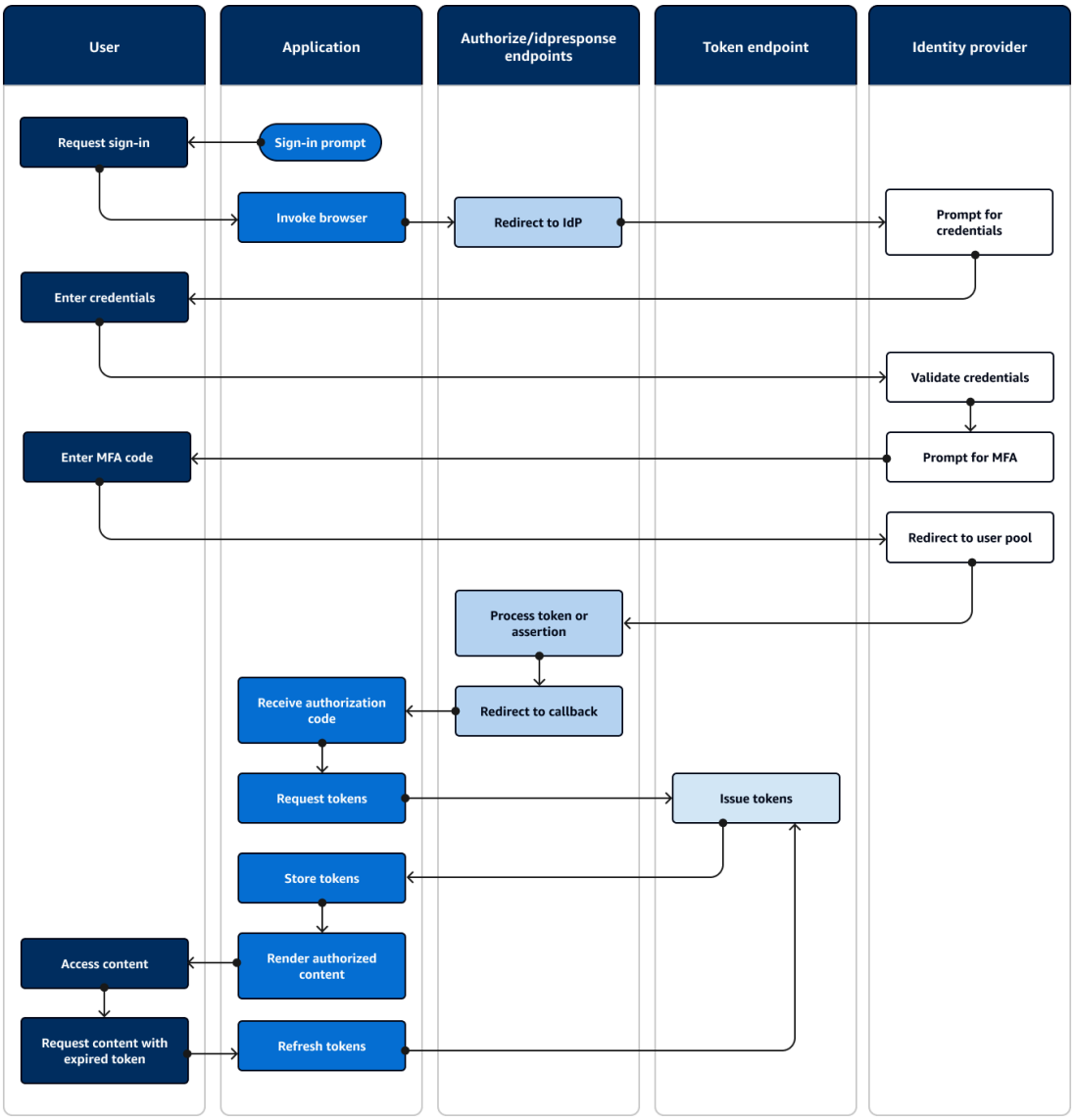
Sign-in con un proveedor de identidad (IdP) externo, o autenticación federada, es un modelo similar al inicio de sesión administrado. Su aplicación es una relación de confianza de OIDC con su grupo de usuarios, mientras que su grupo de usuarios sirve de acceso a un IdP. El IdP puede ser un directorio de usuarios de consumidores, como Facebook o Google, o un directorio empresarial de SAML 2.0 u OIDC, como Azure.

En lugar de utilizar el inicio de sesión administrado en el navegador del usuario, la aplicación invoca un punto de conexión de redireccionamiento en el servidor de autorización del grupo de usuarios. Desde el punto de vista del usuario, este selecciona el botón de inicio de sesión de la aplicación. A continuación, su IdP le pide que inicie sesión. Al igual que ocurre con las autenticaciones del inicio de sesión administrado, una aplicación toma 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. Third-party la autenticación se puede añadir con poco esfuerzo a una aplicación que haya implementado la autenticación de inicio de sesión gestionada. En efecto, el inicio de sesión gestionado y el uso de terceros IdPs generan un resultado de autenticación uniforme a partir de pequeñas variaciones en lo que se invoca en los navegadores de los usuarios.

Al igual que ocurre con las autenticaciones de inicio de sesión administrado, la autenticación federada es la más adecuada para los modelos de autorización avanzados con ámbitos de OAuth 2.0.

En el siguiente diagrama, se muestra un inicio de sesión típico de autenticación federada.



Flujo de autenticación federado

1. El usuario accede a su aplicación.
2. Selecciona un enlace para iniciar sesión.
3. La aplicación dirige al usuario a una petición de inicio de sesión con su IdP.
4. Introduce su nombre de usuario y contraseña.
5. El IdP valida las credenciales del usuario y determina si el usuario ha activado la autenticación multifactor (MFA).
6. El IdP solicita al usuario que introduzca un código de MFA.
7. El usuario introduce el código de la MFA.

8. El IdP redirige al usuario al grupo de usuarios con una respuesta de SAML o un código de autorización.
9. Si el usuario ha conseguido 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 toma el código de autorización del parámetro de solicitud de URL que el grupo de usuarios ha añadido a la URL de devolución de llamada.
11. La aplicación solicita tokens con el código de autorización.
12. El punto de conexión del token devuelve los JWT a la aplicación.
13. La aplicación descodifica, valida y guarda 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 caduca 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 de conexión del token con el token de actualización.

Variantes y personalización

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

Recursos relacionados

- Inicio de sesión en el grupo de usuarios con proveedores de identidad externos
- Ámbitos, M2M y servidores de recursos
- Referencia de los puntos de conexión de grupos de usuarios y del inicio de sesión administrado

Autenticación del grupo de identidades

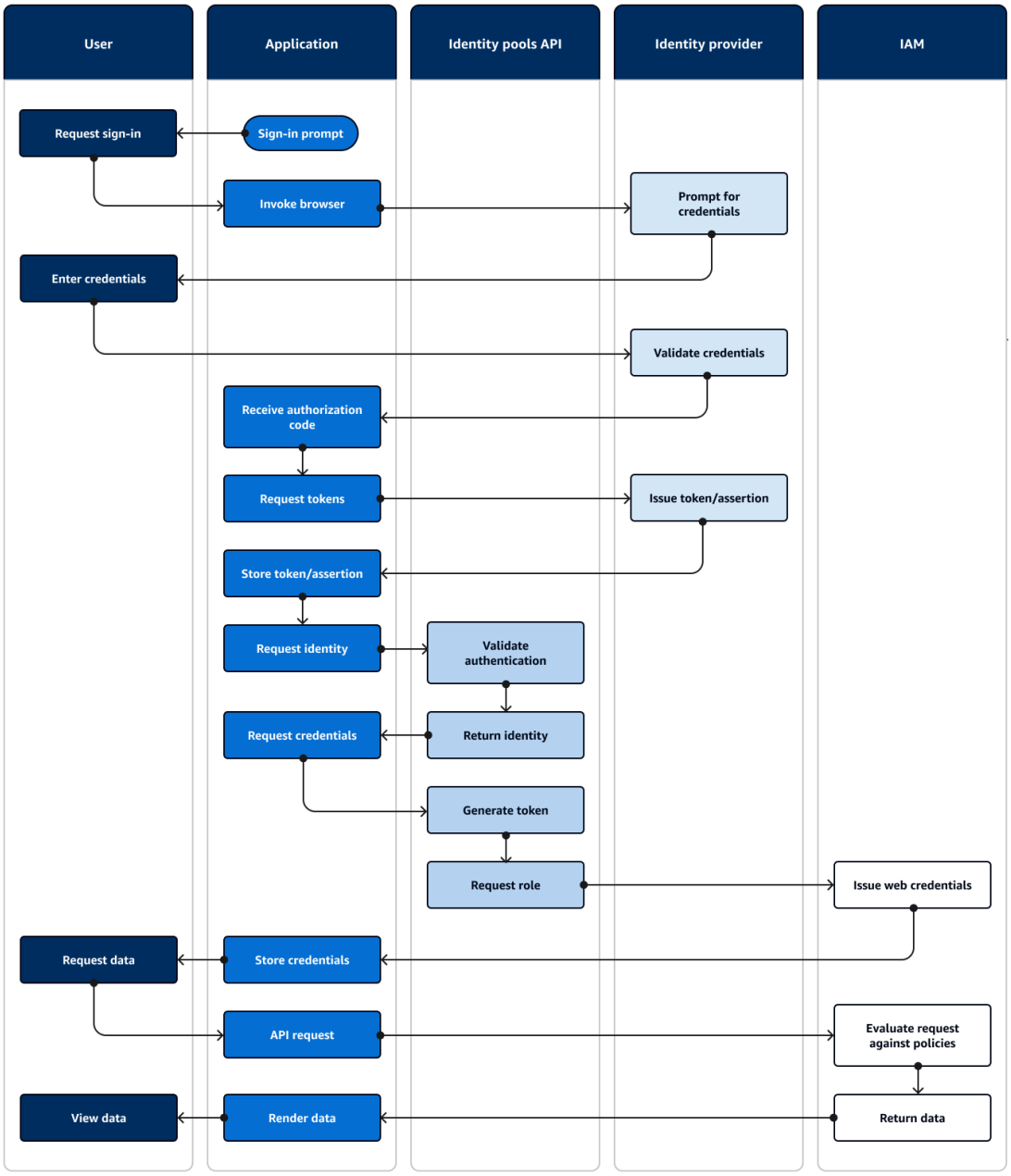
Un grupo de identidades es un componente de la aplicación que se diferencia del grupo de usuarios por la función, el espacio de nombres de la API y el modelo de SDK. Mientras que los grupos de usuarios ofrecen autenticación y autorización basadas en fichas, los grupos de identidades ofrecen autorización 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 las fuentes de identidad de SAML, OIDC, sociales, de desarrolladores e invitadas 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 de grupos 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.

En el siguiente diagrama, se ilustra un inicio de sesión típico de autenticación de grupo de identidades con un IdP.



Flujo de autenticación del grupo de identidades

1. El usuario accede a su aplicación.
2. Selecciona un enlace para iniciar sesión.
3. La aplicación dirige al usuario a una petición de inicio de sesión con su IdP.
4. Introduce 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 de SAML o un código de autorización.
7. Si el usuario ha pasado un código de autorización, la aplicación intercambia el código por los tokens de IdP.
8. La aplicación descodifica, valida y guarda o almacena en caché la aserción o los JWT del usuario.
9. La aplicación invoca el método que realiza una solicitud de GetIdAPI. Transmite el token o la aserción del usuario y solicita un ID de identidad.
10. El grupo de identidades valida el token o la aserción con respecto a los proveedores de identidades configurados.
11. El grupo de identidades devuelve un ID de identidad.
12. La aplicación invoca el método que realiza una solicitud a la GetCredentialsForIdentityAPI. Transmite la aserción o el token del usuario y solicita un rol de IAM.
13. El grupo de identidades genera un nuevo JWT. El nuevo JWT contiene notificaciones que solicitan un rol 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 del 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 un rol de IAM.
15. La aplicación almacena las credenciales de sesión.
16. El usuario realiza una acción en la aplicación que requiere recursos con acceso protegido en AWS.
17. La aplicación aplica las credenciales temporales como firmas a las solicitudes de API para lo necesario Servicios de AWS.
18. IAM evalúa las políticas asociadas al rol en las credenciales. Las compara con la solicitud.
19. Servicio de AWS Devuelve los datos solicitados.
20. La aplicación representa los datos en la interfaz de usuario.
21. El 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 de emisión token/assertion. La autenticación de desarrollador sustituye todos los pasos previos a Solicitar identidad con una solicitud firmada

mediante las credenciales del desarrollador. La autenticación de invitado también pasa directamente a Solicitar identidad, no valida la autenticación y devuelve las credenciales para un rol de IAM de acceso limitado.

Recursos relacionados

- Grupos de identidades de Amazon Cognito
- Roles de IAM de usuario
- Flujo de autenticación de grupos de identidades

Uso de este servicio con un AWS SDK

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 para C++	AWS SDK para C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell ejemplos de código
AWS SDK para Python v4	AWS Ejemplos de código de SDK para Python v4

Documentación de SDK	Ejemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para 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.

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 API aceptan a los usuarios como `Principal` y toman decisiones de autorización de `Action` en el `Resource` al que desean acceder. El código `Context` personalizado adicional 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 notificación `token_use` que figura en su token coincide con los parámetros que ha pasado a `IsAuthorizedWithToken`. La notificación `token_use` debe ser `access` si la ha pasado al parámetro `accessToken` y `id` si la ha pasado al parámetro `identityToken`.
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 la solicitud con `Allow` si coincide con una acción permitida y no con una acción `Deny` 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, agregar o eliminar las notificaciones de usuario que desea presentar a Verified Permissions, personalice el contenido de los tokens de acceso e identidad con un Desencadenador 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.

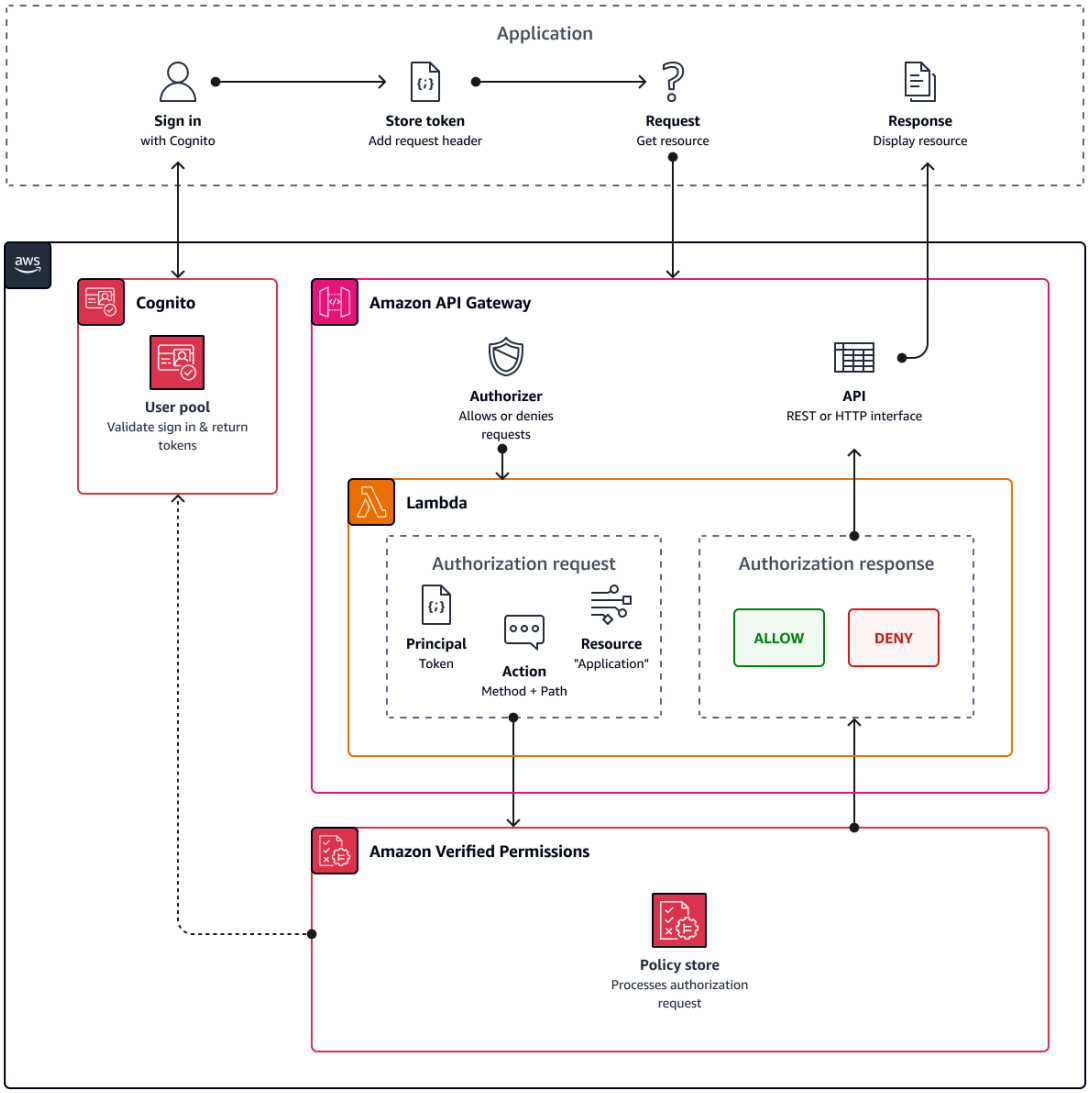
Recursos adicionales

- Mapping Amazon Cognito tokens to Verified Permissions schema
- Authorize API Gateway APIs using Amazon Verified Permissions and Amazon Cognito
- Taller: Autenticación y autorización con Amazon Cognito y Verified Permissions

Autorización de API con Verified Permissions

Los tokens de ID o de acceso pueden autorizar las solicitudes a las API de REST de Amazon API Gateway de backend con Verified Permissions. Puede crear un almacén de políticas con enlaces inmediatos a su grupo de usuarios y a su API. Con la opción de inicio Configuración con API Gateway y un origen de identidades, Verified Permissions añade un origen de identidad del grupo de usuarios al almacén de políticas y un autorizador de Lambda a la API. Cuando la aplicación pasa un token portador del grupo de usuarios a la API, el autorizador de Lambda invoca Verified Permissions. El autorizador transfiere el token como entidad principal y la ruta y el método de solicitud como acción.

El siguiente diagrama ilustra el flujo de autorización de una API de API Gateway con Verified Permissions. Para obtener un desglose detallado, consulta las tiendas API-linked de políticas en la Guía del usuario de permisos verificados de Amazon.



Verified Permissions estructura la autorización de la API en torno a grupos de usuarios. Debido a que los tokens de ID y de acceso incluyen una notificación `cognito:groups`, su almacén de políticas puede administrar el control de acceso basado en roles (RBAC) para las API en diversos contextos de la aplicación.

Elección de la configuración del almacén de políticas

Al configurar un origen de identidad en un almacén de políticas, debe elegir si desea procesar los tokens de acceso o de ID. Esta decisión es importante para el funcionamiento del motor de políticas. Los tokens de ID contienen atributos de usuario. Los tokens de acceso contienen información sobre el control de acceso de los usuarios: ámbitos de OAuth. Aunque ambos tipos de token contienen información sobre la pertenencia a un grupo, generalmente recomendamos el token de

acceso para el RBAC con un almacén de políticas de Verified Permissions. El token de acceso precisa la pertenencia a un grupo con ámbitos que pueden contribuir a la decisión de autorización. Las notificaciones de un token de acceso pasan a formar parte del contexto de la solicitud de autorización.

También debe configurar los tipos de entidad de usuario y grupo al configurar un grupo de usuarios como origen de identidad. Los tipos de entidad son identificadores de entidades principales, de acciones y de recursos a los que puede hacer referencia en las políticas de Verified Permissions. 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 de entidades 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. Cuando configuras un almacén de API-linked políticas o sigues la configuración guiada en la consola de permisos verificados, tu almacén de políticas tiene automáticamente esta relación padre-miembro.

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

También puede configurar un almacén de políticas para que acepte tokens con una notificación `aud` o `client_id` que coincidan con una lista de clientes de aplicación aceptables que haya proporcionado.

Ejemplo de política para una autorización de API basada en el rol

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 decisión `Allow` a la solicitud de autorización de su aplicación cuando:

1. La aplicación ha pasado un token de ID o de acceso en un encabezado `Authorization` como token de portador.
2. La aplicación ha pasado un token con una notificación `cognito:groups` con la cadena `MyGroup`.
3. La aplicación ha presentado una solicitud `HTTP GET` 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 Verified Permissions en condiciones distintas de las solicitudes de API. Puede enviar cualquier decisión de control de acceso de su aplicación a su almacén de políticas. Puede, por ejemplo, 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 de entidad principal para un usuario de un grupo de usuarios con un ID `us-east-1_Example` con sub, o ID 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 Verified Permissions, utilice el siguiente formato:

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

Attribute-based control de acceso

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.

Referencia de los puntos de conexión de grupos de usuarios y del inicio de sesión administrado

Amazon Cognito dispone de dos modelos de autenticación de grupos de usuarios: la API de grupos de usuarios y el servidor de autorización OAuth 2.0. Utilice la API cuando desee recuperar los tokens de OpenID Connect (OIDC) con un AWS SDK en el back-end de su aplicación. Utilice el servidor de autorización cuando desee implementar su grupo de usuarios como proveedor OIDC. El servidor de autorización añade características como el inicio de sesión federado, la autorización API y M2M con ámbitos de tokens de acceso y el inicio de sesión administrado. Ambos modelos se pueden usar juntos o por separado, configurados por grupo de usuarios o por cliente de aplicación. Esta sección sirve de referencia para implementar el modelo de OIDC. Para obtener más información sobre los dos modelos de autenticación, consulte Descripción de la autenticación mediante API, OIDC y páginas de inicio de sesión administrado.

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 el inicio de sesión administrado, donde sus usuarios pueden registrarse e iniciar sesión (Punto de conexión Login) y cerrar sesión (Punto de conexión Logout). Para obtener más información acerca de estos recursos, consulte Inicio de sesión administrado de grupos de usuarios.

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 en el inicio de sesión administrado interactivo Punto de conexión Login o el Autorizar punto de conexión de OIDC. El punto de conexión de autorización redirige a los usuarios a las páginas de inicio de sesión administrado o a la página de inicio de sesión del 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 del inicio de sesión gestionado, 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 Gestor de certificados .

Para obtener una lista de los puntos de enlace de servicio de la API de grupos de usuarios Región de AWS, consulta los puntos de enlace de servicio en la AWS Referencia general.

Temas

- User-interactive inicio de sesión gestionado y puntos de enlace de interfaz de usuario alojada clásicos
- Puntos de conexión de los proveedores de identidades y de la relación de confianza
- Concesiones de OAuth 2.0
- Uso de la PKCE en las concesiones de códigos de autorización
- Respuestas de error de federación y de inicio de sesión administrado

User-interactive inicio de sesión gestionado y puntos de enlace de interfaz de usuario alojada clásicos

Amazon Cognito activa los puntos de conexión del inicio de sesión administrado en esta sección cuando usted añade un dominio al grupo de usuarios. Son páginas web en las que los usuarios pueden ejecutar 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.

Las páginas web que componen el inicio de sesión administrado son una aplicación web frontend para sesiones de usuario interactivas con sus clientes. Su aplicación debe invocar el inicio de sesión

administrado 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 Puntos de conexión de los proveedores de identidades y de la relación de confianza que devuelven una respuesta JSON pueden consultarse directamente en su código de aplicación. El Autorizar punto de conexión redirige al inicio de sesión administrado o a una página de inicio de sesión del IdP, y también debe abrirse en los navegadores de los usuarios.

Todos los puntos de conexión del grupo de usuarios aceptan el tráfico de las direcciones IP de origen IPv4 e IPv6.

En los temas de esta guía se describen en detalle el inicio de sesión gestionado más utilizado y los puntos finales de la interfaz de usuario alojada clásica. La diferencia entre el inicio de sesión administrado y la interfaz de usuario alojada es visible, no funcional. Con la excepción de `/passkeys/add`, todas las rutas se comparten entre las dos versiones de la creación de marca del inicio de sesión administrado.

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 en el inicio de sesión administrado

URL del punto de conexión	Description (Descripción)	Cómo se accede
https:///login *Your user pool domain*	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:///cerrar sesión *Your user pool domain*	Cierra la sesión de los usuarios del grupo de usuarios.	Enlace directo. Consulte Punto de conexión Logout.
https:///ConfirmUser *Your user pool domain*	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.

URL del punto de conexión	Description (Descripción)	Cómo se accede
https:///registrarse *Your user pool domain*	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:///confirmar *Your user pool domain*	Cuando el grupo de usuarios envía un código de confirmación a un usuario que se haya registrado, se lo pedirá al usuario.	Redirect-only de`/signup`.
https://*Your user pool domain*/Olvidé mi contraseña	Solicita al usuario su nombre de usuario y envía un código para restablecer la 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	Description (Descripción)	Cómo se accede
https://////ConfirmForgotPassword *Your user pool domain*	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). Una vez que el usuario haya enviado un código de restablecimiento válido y una nueva contraseña, lo `/confirmforgotPassword` redireccionará `/login` a para que inicie sesión con su nueva contraseña.	Redirect-only desde. `/forgotPassword`
https://*Your user pool domain*/resendcode	Envía un nuevo código de confirmación a un usuario que se ha registrado en el grupo de usuarios.	Redirect-only de Enviar un nuevo enlace de código a. `/confirm`
https://*Your user pool domain*/passkeys/add	Registra una nueva clave de acceso. Solo está disponible en un inicio de sesión administrado.	• En el flujo de registro, tras la confirmación en los clientes de aplicación que admiten la autenticación con clave de acceso. • Enlace directo con los mismos parámetros que `/oauth2/authorize` .

Temas

- El punto final de inicio de sesión gestionado: /login

- El punto final de inicio de sesión y cierre de sesión gestionado: /logout

El punto final de inicio de sesión gestionado: /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 al inicio de sesión administrado 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 del inicio de sesión administrado. En la aplicación, invoque las páginas de inicio de sesión administrado y de federación 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 la redirecciona a `/login` desde el Autorizar punto de conexión, transmite todos los parámetros que ha proporcionado en la 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: Pedir 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=1example23456789&
                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
```

User-initiated solicitud de inicio de sesión

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.

Step-up solicitud de autenticación

El `/login` punto final acepta los siguientes parámetros de URL para la autenticación progresiva. Estos parámetros tienen el mismo comportamiento que en el punto final de autorización. Para obtener más información sobre los niveles de ACR y los valores de AMR, consulte. the section called “Niveles de autenticación con notificaciones de ACR y AMR”

`acr_values`

Una URL-encoded lista separada por espacios de los URI del nivel de autenticación ACR objetivo, del más alto al más bajo. Amazon Cognito evalúa la lista de izquierda a derecha, aplica el primer nivel válido que el usuario pueda cumplir e ignora los valores que no reconoce. Este parámetro se comporta igual que el parámetro de la API. `TARGET_ACR_VALUES` Requiere el plan de funciones Essentials o Plus.

`max_age`

Un entero no negativo que especifica el número máximo de segundos transcurridos desde la última vez que el usuario se autenticó. Si el tiempo actual negativo `auth_time` es mayor

quemax_age, el usuario debe volver a autenticarse desde cero en lugar de limitarse a acelerar. Este comportamiento es idéntico al parámetro de la MAX_AGE API.

Para ver un ejemplo de solicitud que usa estos parámetros, consulta el punto final de autorización.

max_ageCombínelo acr_values con la exigencia de un nivel de autenticación específico con una autenticación reciente para una operación confidencial.

El comportamiento de un plan de funciones insuficiente difiere del de la API. El /login punto final de inicio de sesión gestionado ignora de forma silenciosa la autenticación normal acr_values y continúa con ella, sin que se produzca ningún error. La API devuelve un. FeatureUnavailableInTierException Para obtener más información, consulte Requisitos del plan de funciones.

El punto final de inicio de sesión y cierre de sesión gestionado: /logout

El punto de conexión /logout es un punto de conexión de redirección. Cierra la sesión del usuario y redirige a una URL de cierre de sesión autorizada para el cliente de la aplicación o al punto de conexión /login. Los parámetros disponibles en una solicitud GET al punto de conexión /logout se adaptan a los casos de uso del inicio de sesión administrado en Amazon Cognito.

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 inicio de sesión administrado en los navegadores de sus usuarios.

A fin de redirigir al usuario al inicio de sesión administrado 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.

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. Cuando el IdP vuelve a redirigir

al usuario a `saml2/logout`, Amazon Cognito responde con otra redirección a `redirect_uri` o a `logout_uri` desde la solicitud. Para obtener más información, consulte Cierre de sesión de usuarios de SAML con un cierre de sesión único.

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 un usuario en un IdP externo, debe dirigirlo 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 Application-specific configuración con clientes de aplicaciones.

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 Application-specific configuración con clientes de aplicaciones.

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 tu solicitud al `/logout` punto final, establece el valor del parámetro logout_uri en la página de inicio de sesión. URL-encoded

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 en la solicitud al punto

de conexión se incluyen los parámetros logout_uri y redirect_uri, `/logout` Amazon Cognito utilizará exclusivamente el parámetro logout_uri, anulando el parámetro redirect_uri.

nonce

(Opcional) Valor aleatorio que puede 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.

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 a un punto de conexión `/login` para volver a autenticar y pasar tokens a su aplicación, añada un parámetro redirect_uri. Si los parámetros logout_uri y redirect_uri están ambos incluidos en la solicitud al punto de conexión de `/logout`, Amazon Cognito anulará el parámetro redirect_uri y procesará únicamente el parámetro logout_uri.

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 state a una solicitud, Amazon Cognito devuelve su valor a la aplicación cuando el punto de conexión `/oauth2/logout` redirige al usuario.

Agregue este valor a sus solicitudes de protección contra ataques CSRF.

No puede establecer el valor de URL-encoded un parámetro `state` en una cadena JSON. Para pasar una cadena que coincida con este formato en un parámetro `state`, codifique la cadena en base64 y luego descodifíquela en la 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.

Solicitudes de ejemplo

Ejemplo: Cerrar sesión y redirigir al usuario al cliente

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 pedir 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 del inicio de sesión administrado. El punto de conexión de cierre de sesión anexa los parámetros de la solicitud original al destino de redireccionamiento.

Los parámetros adicionales que añada a la solicitud de cierre de sesión deben estar en la lista de Parámetros de solicitud. Por ejemplo, el punto de conexión de cierre de sesión no admite la redirección automática de IdP con parámetros `identity_provider` o `idp_identifier`. 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 posterior al inicio de sesión por la que es recomendable pasar hacia el punto de conexión de autorización.

A continuación mostramos 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 a `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
```

Puntos de conexión de los proveedores de identidades y de la relación de confianza

Los puntos de enlace de federación son puntos de enlace de grupos de usuarios que cumplen un propósito para uno de los estándares de autenticación utilizados por los grupos de usuarios. Contienen direcciones URL de ACS de SAML, puntos de conexión de detección de OIDC y puntos de conexión de servicio para roles de grupos de usuarios, tanto de proveedor de identidades como de la relación de confianza. Los puntos finales de la federación inician los flujos de autenticación, reciben una prueba de IdPs autenticación y emiten tokens a los clientes. Interactúan con IdPs las aplicaciones y los administradores, pero no con los usuarios.

Los temas que se abordan en la páginas completas posteriores a esta página contienen detalles sobre los puntos de conexión de los proveedores OAuth 2.0 y OIDC que pasan a estar disponibles cuando añade un dominio a un grupo de usuarios. En el siguiente gráfico, se muestra una lista de todos los puntos de conexión de federación.

Algunos ejemplos de dominios de grupos de usuarios:

1. Dominio de prefijo: `mydomain.auth.us-east-1.amazoncognito.com`
2. Dominio personalizado: `auth.example.com`

Puntos de conexión de federación de grupo de usuarios

URL del punto de conexión	Description (Descripción)	Cómo se accede
https://*Your user pool domain*/oauth2/authorize	Redirige un usuario al inicio de sesión administrado o a iniciar sesión con el IdP.	Se invoca en el navegador del cliente para iniciar la autentica ción del usuario. Consulte Autorizar punto de conexión.
https://*Your user pool domain*/oauth2/token	Devuelve los tokens en función de un código de autorización o de una solicitud de credenciales del cliente.	La aplicación la solicita para recuperar los tokens. Consulte Punto de conexión de token.
https://*Your user pool domain*/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 la solicita para recuperar el perfil de usuario. Consulte El punto de conexión userInfo.
https://*Your user pool domain*/oauth2/revoke	Revoca un token de actualiza ción y los tokens de acceso asociados.	La aplicación la solicita para revocar un token. Consulte Revocación de puntos de conexión.
https://cognito-id p.*Regionyour user pool ID*.amazonaws.com/ /.well- - configuración known/openid	Un directorio de la arquitect ura OIDC de su grupo de usuarios. [1]	La aplicación la solicita para localizar los metadatos del emisor del grupo de usuarios.
https://cognito-id p.*Regionyour user pool ID*.amazonaws.com/ /.well-.j son known/jwks	Claves públicas que puede usar para validar los tokens de Amazon Cognito. [2]	La aplicación la solicita para verificar los JWT.
https://*Your user pool domain*/oauth2/idpresponse	Los proveedores de identidad es social deben redirigir a los usuarios a este punto de conexión con un código de	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	Description (Descripción)	Cómo se accede
	autorización. Amazon Cognito canjea el código por un token cuando autentica al usuario federado.	
https://*Your user pool domain*/saml2/idpresponse	La URL de Assertion Consumer Response (ACS) para la integración con los proveedores de identidades de SAML 2.0.	Redirigido desde el IdP de SAML 2.0 como URL de ACS o punto de origen para el inicio de sesión. IdP-initiated [3]
https:///*Your user pool domain*saml2/logout	La URL de cierre de sesión único (SLO) para la integración con los proveedores de identidades SAML 2.0.	Se redirige desde el IdP SAML 2.0 como URL de cierre de sesión único (SLO). Solo acepta enlaces de tipo POST.

[1] El documento `openid-configuration` puede actualizarse en cualquier momento con información adicional que permita que el punto de conexión cumpla con las especificaciones de OIDC y OAuth2.

[2] El archivo `jwks.json` JSON se puede actualizar en cualquier momento con nuevas claves públicas de firma de fichas.

[3] Para obtener más información sobre el inicio de sesión con IdP-initiated SAML, consulte. Implemente el inicio de IdP-initiated sesión mediante SAML

Para obtener más información acerca de los estándares OpenID Connect y OAuth, consulte OpenID Connect 1.0 y OAuth 2.0.

Grupos de usuarios de Amazon Cognito como emisor de OIDC

Los grupos de usuarios de Amazon Cognito funcionan como proveedores de identidad de OpenID Connect (OIDC) y actúan como un emisor que las bibliotecas de aplicaciones pueden utilizar para la federación de OIDC. Las bibliotecas de aplicaciones para la federación de OIDC pueden hacer referencia a las dos rutas diferentes, tal como se explica a continuación, como punto final de detección automática. Este punto final proporciona acceso al conjunto de claves web JSON

(JWKS) `/.well-known/jwks.json` y a los metadatos de descubrimiento de OIDC en`/.well-known/openid-configuration`, donde las aplicaciones pueden descubrir los puntos finales de autorización, token y UserInfo.

Las aplicaciones que admiten la detección automática del OIDC pueden configurarse automáticamente consultando estos puntos finales conocidos. En el caso de las aplicaciones que no admiten la detección automática, puede codificar la aplicación con los puntos finales específicos del OIDC que se enumeran en la sección anterior.

Tipos de emisores de grupos de usuarios

Emisor original

La configuración de emisor predeterminada actual para los grupos de usuarios. La URL del emisor está alojada en la región del grupo de usuarios y proporciona los puntos de conexión OIDC específicos de esa región.

Los emisores originales adoptan el formato. `https://cognito-idp.`*`us-east-1`*`.amazonaws.com/`*`us-east-1_EXAMPLE`*

Emisor actualizado

Se recomienda para todos los grupos de usuarios, incluida la replicación multirregional. Los emisores actualizados alojan el mismo contenido de JWKS en varias regiones, lo que mejora la resiliencia y la eficiencia.

Los emisores actualizados adoptan el formato`https://issuer-cognito-idp.`*`us-east-1`*`.amazonaws.com/`*`us-east-1_EXAMPLE`*, que *`Region`* es el principal Región de AWS de su grupo de usuarios.

> **Note**
>
> El tipo de emisor actualizado no es compatible con la integración nativa de Amazon Cognito para la autenticación `authenticate-cognito` de Application Load Balancer (la acción). Para usar un emisor actualizado con un Application Load Balancer, configure la integración genérica de OIDC (la acción) en su lugar y proporcione `authenticate-oidc` los puntos finales de OIDC de su grupo de usuarios. Para obtener más información, consulte Autenticar usuarios mediante un Equilibrador de carga de aplicación.

Prepárese para los cambios en el documento de descubrimiento

El `openid-configuration` documento es un documento de descubrimiento estándar de OpenID Connect. Amazon Cognito puede añadirle entradas de metadatos a lo largo del tiempo para que cumpla con las especificaciones de OpenID Connect y OAuth 2.0 y para mejorar la compatibilidad con las bibliotecas cliente de OIDC. Estas incorporaciones son compatibles con versiones anteriores: Amazon Cognito añade nuevos miembros, pero no cambia el nombre, el tipo de miembros que la aplicación ya lee ni los elimina. Escriba a su cliente para que siga funcionando a medida que el documento vaya creciendo.

Ignora a los miembros que no reconozcas

> Como exige la especificación OpenID Connect Discovery, trate los miembros de metadatos que no comprenda como opcionales. No rechace, falle ni arroje un error cuando el documento contenga un miembro que su aplicación no utilice.

No codifique de forma rígida el conjunto ni el orden de los miembros

> Analice el documento como un objeto JSON general en lugar de como una estructura fija. Los miembros pueden aparecer en cualquier orden y el conjunto de miembros puede cambiar de una solicitud a otra y a lo largo del tiempo. Diríjase a cada miembro por su nombre y léalos según el tipo que la especificación defina para ellos.

Compare los clientes estrictos con el documento actual

> Si su cliente valida estrictamente el documento de detección, busque el documento actual en el `/.well-known/openid-configuration` punto final de su grupo de usuarios y confirme que su cliente lo acepta antes de que dependa de un conjunto específico de miembros.

Los tokens de Amazon Cognito siguen el mismo enfoque aditivo y compatible con versiones posteriores. Para obtener más información sobre cómo analizar las afirmaciones en un identificador o token de acceso, consulte. Analice los tokens del grupo de usuarios de forma segura

Temas

- El punto de conexión de redireccionamiento y autorización
- El punto final del token
- El punto de conexión de los atributos de usuario
- El punto de conexión de revocación del token
- El punto de conexión de aserción del IdP SAML

El punto de conexión de redireccionamiento y autorización

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 al inicio de sesión administrado 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 a la que desea que se le redirija.
- 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.

`OBTÉN/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 de cliente de 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 dirección 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 su aplicación agrega un parámetro state a una solicitud, Amazon Cognito devuelve su valor a la aplicación cuando el punto de conexión `/oauth2/authorize` redirige al usuario.

Agregue este valor a sus solicitudes de protección contra ataques CSRF.

No se puede establecer el valor de un `state` parámetro en una cadena URL-encoded JSON. Para pasar una cadena que coincida con este formato en un parámetro `state`, codifique la cadena en base64 y luego descodifíquela en la aplicación.

`identity_provider`

Opcional.

Agregue este parámetro para omitir el inicio de sesión administrado y redirigir al usuario a la página de inicio de sesión del proveedor. El valor delidentity_provideres el nombre del proveedor de identidad (IdP) tal como aparece en el grupo de usuarios.

- En el caso de los proveedores de redes sociales, puede usar los valores de identity_provider `Facebook`, `Google`, `LoginWithAmazon` y `SignInWithApple`.
- En cuanto a 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 redirigir a un proveedor con un nombre alternativo para el nombre identity_provider. Puede introducir los identificadores de SAML 2.0 y OIDC en el menú IdPs de proveedores sociales y externos de la consola de Amazon Cognito.

`scope`

Opcional.

Puede ser una combinación de cualquier ámbito reservado por el sistema o ámbitos personalizados asociados 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 ha utilizado 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 de prueba de intercambio de códigos clave (PKCE) que ha generado a partir de `code_verifier`. Para obtener más información, consulte Uso de la PKCE en las concesiones de códigos de autorización.

Obligatorio solo cuando se especifica un parámetro `code_challenge_method`.

`nonce`

Opcional.

Valor aleatorio que puede agregar 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.

`lang`

Opcional.

El idioma en el que desea mostrar las páginas interactivas para el usuario. Las páginas de inicio de sesión administrado se pueden localizar, pero las páginas de la interfaz de usuario

alojada (clásicas) no. Para obtener más información, consulte Localización de inicio de sesión administrado.

`login_hint`

Opcional.

La petición de nombre de usuario que desea pasar al servidor de autorización. Puede recopilar el nombre de usuario, la dirección de correo electrónico o el número de teléfono del usuario y permitir que el proveedor de destino rellene previamente el nombre de inicio de sesión del usuario. Cuando envía un parámetro `login_hint` y ningún parámetro `idp_identifier` ni `identity_provider` al punto de conexión `oauth2/authorize`, el inicio de sesión administrado rellena el campo del nombre de usuario con el valor de la sugerencia. También puede pasar este parámetro al Punto de conexión Login y rellenar automáticamente el valor del nombre de usuario.

Cuando su solicitud de autorización invoca una redirección a OIDC, Amazon IdPs Cognito añade un `login_hint` parámetro a la solicitud a ese autorizador externo. No puedes reenviar las sugerencias de inicio de sesión a SAML, Apple, Login With Amazon, Google o Facebook (Meta) IdPs.

`prompt`

Opcional.

Parámetro de OIDC que controla el comportamiento de autenticación de las sesiones existentes. Disponible solo en la versión de creación de marca de inicio de sesión administrado, no en la interfaz de usuario alojada clásica. Para obtener más información sobre la especificación OIDC, consulte Solicitud de autorización. Los valores `none` y `login` tienen un efecto en el comportamiento de autenticación del grupo de usuarios.

Amazon Cognito reenvía todos los valores `prompt` excepto `none` a usted IdPs cuando los usuarios seleccionan la autenticación con proveedores externos. Esto ocurre cuando la URL a la que acceden los usuarios incluye un `idp_identifier` parámetro `identity_provider` o, o cuando el servidor de autorización los redirige a Punto de conexión Login y ellos seleccionan un IdP entre los botones disponibles.

Valores de parámetros de petición

`prompt=none`

Amazon Cognito continúa la autenticación de forma silenciosa para los usuarios que tienen una sesión autenticada válida. Con esta petición, los usuarios pueden autenticarse de forma silenciosa entre diferentes clientes de aplicaciones de su grupo de usuarios. Si el usuario aún no se ha autenticado, el servidor de autorización devuelve un error `login_required`.

`prompt=login`

Amazon Cognito requiere que los usuarios se vuelvan a autenticar aunque tengan una sesión existente. Envíe este valor cuando desee volver a verificar la identidad del usuario. Los usuarios autenticados que tienen una sesión existente pueden volver a iniciar sesión sin invalidar esa sesión. Cuando un usuario que tiene una sesión existente vuelve a iniciar sesión, Amazon Cognito le asigna una nueva cookie de sesión. Este parámetro también se puede reenviar a su. IdPs IdPslos que aceptan este parámetro también solicitan al usuario un nuevo intento de autenticación.

`prompt=select_account`

Este valor no afecta al inicio de sesión local y debe enviarse en las solicitudes que redirijan a IdPs. Cuando se incluye en la solicitud de autorización, este parámetro añade `prompt=select_account` a la ruta URL del destino de redireccionamiento del IdP. Cuando IdPs admiten este parámetro, solicitan a los usuarios que seleccionen la cuenta con la que desean iniciar sesión.

`prompt=consent`

Este valor no afecta al inicio de sesión local y debe enviarse en las solicitudes que redirijan a IdPs. Cuando se incluye en la solicitud de autorización, este parámetro añade `prompt=consent` a la ruta URL del destino de redireccionamiento del IdP. Cuando IdPs admiten este parámetro, solicitan el consentimiento del usuario antes de volver a redirigirlo a tu grupo de usuarios.

Al omitir el parámetro `prompt` de la solicitud, el inicio de sesión administrado sigue el comportamiento predeterminado: los usuarios deben iniciar sesión a menos que su navegador tenga una cookie de inicio de sesión administrado válida. Puede combinar varios valores para `prompt` con un delimitador de caracteres y espacios, como `prompt=login consent`.

`resource`

Opcional.

El identificador de un recurso que quiere vincular al token de acceso en la reclamación `aud`. Al incluir este parámetro, Amazon Cognito valida que el valor es una URL y establece la audiencia del token de acceso resultante en el recurso solicitado. Puede solicitar un servidor de recursos de grupo de usuarios con un identificador en formato de URL o una URL de su elección. Los valores de este parámetro deben empezar por `https://`, `http://localhost` o un esquema de URL personalizado, como `myapp://`.

La vinculación de recursos se define en RFC 8707. Para obtener más información acerca de los servidores de recursos y la vinculación de recursos, consulte Vinculación de recursos.

Ejemplo: concesión de código de autorización

A continuación mostramos un ejemplo de solicitud de concesión de código de autorización.

La solicitud siguiente inicia una sesión para recuperar un código de autorización que el usuario pasa a la aplicación en el destino de `redirect_uri`. En dicha 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
```

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
```

Ejemplo: concesión de código de autorización con PKCE

Este ejemplo de flujo realiza una concesión de código de autorización con PKCE.

Esta solicitud agrega un parámetro `code_challenge`. Para intercambiar un código por un token, debe incluir el parámetro `code_verifier` en la solicitud para el punto de conexión `/oauth2/token`.

```
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...
```

El servidor de autorización devuelve la redirección a la aplicación con el estado y el código de autorización. Su aplicación procesa el código de autorización y lo cambia por tokens.

```
HTTP/1.1 302 Found
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Ejemplo: requerir una nueva autenticación con prompt=login

La siguiente solicitud agrega un parámetro `prompt=login` que requiere que el usuario se autentique nuevamente, aunque ya tenga una sesión existente.

```
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&
prompt=login
```

El servidor de autorización redirige al punto de conexión de inicio de sesión, por lo que es necesario volver a autenticarse.

```
HTTP/1.1 302 Found Location: https://mydomain.auth.us-east-1.amazoncognito.com/
login?response_type=code&client_id=1example23456789&redirect_uri=https://
www.example.com&state=abcdefg&scope=openid+profile
+aws.cognito.signin.user.admin&prompt=login
```

Ejemplo: autenticación silenciosa con prompt=none

La siguiente solicitud agrega un parámetro `prompt=none` que compruebe silenciosamente si el usuario tiene una sesión válida.

```
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&
prompt=none
```

Cuando no existe una sesión válida, el servidor de autorización devuelve un error al URI de redireccionamiento

```
HTTP/1.1 302 Found Location: https://www.example.com?error=login_required&state=abcdefg
```

Cuando existe una sesión válida, el servidor de autorización devuelve un código de autorización.

```
HTTP/1.1 302 Found Location: https://www.example.com?
code=AUTHORIZATION_CODE&state=abcdefg
```

Ejemplo: concesión de código de autorización con vinculación de recursos

La siguiente solicitud agrega un parámetro `resource` para vincular el token de acceso a un servidor de recursos específico. El token de acceso resultante crea las condiciones para que la API de destino valide que se trata de la audiencia prevista de la solicitud del usuario autenticado.

```
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=solar-system-data-api.example.com/asteroids.add&
resource=https://solar-system-data-api.example.com
```

El servidor de autorización devuelve un código de autorización que da como resultado un token de acceso con una reclamación `aud` de `https://solar-system-data-api.example.com`.

```
HTTP/1.1 302 Found Location: https://www.example.com?
code=AUTHORIZATION_CODE&state=abcdefg
```

Ejemplo: concesión simbólica (implícita) sin el ámbito de openid

Este flujo de ejemplo genera una concesión implícita y devuelve JWT directamente a la sesión del usuario.

La solicitud sirve para obtener una concesión implícita del servidor de autorización. Solicita ámbitos en el token de acceso que autoricen las operaciones de autoservicio del perfil de usuario.

```
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
```

El servidor de autorización redirige de vuelta a su aplicación solo con un 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.

```
HTTP/1.1 302 Found
Location: https://example.com/
callback#access_token=eyJra456defEXAMPLE&token_type=bearer&expires_in=3600&state=STATE
```

Ejemplo: concesión simbólica (implícita) con ámbito openid

Este flujo de ejemplo genera una concesión implícita y devuelve tokens al navegador del usuario.

La solicitud sirve para obtener una concesión implícita del servidor de autorización. Solicita ámbitos en el token de acceso que autorizan el acceso a los atributos del usuario y a las operaciones 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
```

El servidor de autorización redirige a la aplicación con el token de acceso y el token de ID (porque se ha incluido el ámbito `openid`):

```
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 la solicitud. Las solicitudes negativas vienen con un código de error HTTP y una descripción que puede utilizar para corregir los parámetros de la solicitud. A continuación se proporcionan ejemplos de respuestas negativas.

- Si `client_id` y `redirect_uri` son válidos, pero los parámetros de solicitud no tienen el formato correcto, el servidor de autenticación redirige el error al `redirect_uri` del cliente y añade un mensaje de error en un parámetro URL. A continuación se proporcionan ejemplos de formatos incorrectos.
  - La solicitud no incluye un parámetro `response_type`.
  - La solicitud de autorización ha proporcionado un parámetro `code_challenge`, pero no un parámetro `code_challenge_method`.
  - El valor del parámetro `code_challenge_method` no es `S256`.

  A continuación mostramos un ejemplo de respuesta con formato incorrecto.

  ```
  HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_request
  ```

- Si el cliente solicita `code` o `token` en `response_type`, pero no tiene permiso para estas solicitudes, el servidor de autorización de Amazon Cognito devuelve `unauthorized_client` al `redirect_uri` del cliente, tal y como se indica a continuación:

  ```
  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 algún error inesperado en el servidor, el servidor de autenticación devuelve `server_error` al `redirect_uri` del cliente. No debe mostrarse el error HTTP 500 en el navegador del usuario porque este error no se envía al cliente. 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 durante la llamada al punto de conexión `jwks_uri` para validar el token de ID, el servidor de autenticación redirige el error al `redirect_uri` del cliente, tal y como se indica a continuación:

```
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 con un tercero, es posible que los proveedores IdPs devuelvan respuestas de error. Esto puede deberse a errores de configuración u 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
    ```

Ejemplo: solicitud de autenticación Step-up

Amazon Cognito admite la autenticación escalonada con el inicio de sesión gestionado. No se admite en la interfaz de usuario alojada. Si su aplicación se basa en ACR y AMR, utilice el inicio de sesión gestionado.

Para obtener más información sobre los niveles de ACR y los valores de AMR, consulte. the section called “Niveles de autenticación con notificaciones de ACR y AMR”

El `/oauth2/authorize` punto final acepta los siguientes parámetros de URL para aumentar la autenticación.

`acr_values`

Una URL-encoded lista separada por espacios de los URI de nivel de autenticación ACR de destino, del más alto al más bajo. Amazon Cognito evalúa la lista de izquierda a derecha, aplica el primer nivel válido que el usuario pueda cumplir e ignora los valores que no reconoce. Este parámetro se comporta igual que el parámetro de la API. `TARGET_ACR_VALUES` Requiere el plan de funciones Essentials o Plus.

`max_age`

Un entero no negativo que especifica el número máximo de segundos transcurridos desde la última vez que el usuario se autenticó. Si el tiempo actual negativo `auth_time` es mayor que`max_age`, el usuario debe volver a autenticarse desde cero en lugar de limitarse a acelerar. Este comportamiento es idéntico al parámetro de la `MAX_AGE` API.

`max_age`Combínelo `acr_values` con la exigencia de un nivel de autenticación específico con una autenticación reciente para una operación confidencial.

En la siguiente `/oauth2/authorize` solicitud de ejemplo, se pide a Amazon Cognito que pruebe primero el nivel 4. Si el usuario no puede cumplir con el nivel 4, por ejemplo, porque no ha configurado el TOTP, Amazon Cognito vuelve al nivel 3, suponiendo que el usuario tenga configurada una contraseña de un solo uso para SMS o correo electrónico. La solicitud también requiere que el usuario se haya autenticado en los últimos 3.600 segundos. El ID de cliente de ejemplo *`1example23456789`* y los demás valores en cursiva son marcadores de posición. Reemplácelos con valores propios.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://YOUR_APP/
redirect_uri&state=STATE&scope=openid+profile
+aws.cognito.signin.user.admin&acr_values=urn%3Acognito%3Aloa%3A4+urn%3Acognito%3Aloa
%3A3&max_age=3600
```

El comportamiento de un plan de funciones insuficiente difiere del de la API. El terminal de autorización de inicio de sesión gestionado ignora de forma silenciosa `acr_values` y continúa con la autenticación normal, sin que se produzca ningún error. La API devuelve un. `FeatureUnavailableInTierException` Para obtener más información, consulte Requisitos del plan de funciones.

El punto final del token

El punto de conexión del token OAuth 2.0 en `/oauth2/token` emite tokens web JSON (JWT) a las aplicaciones que desean completar los flujos de concesión de códigos de autorización y credenciales de cliente. Estos tokens son el resultado final de la autenticación con un grupo de usuarios. Contienen información sobre el usuario (token de ID), su nivel de acceso (token de acceso) y su derecho a conservar la sesión en la que ha iniciado sesión (token de actualización). Las bibliotecas de relaciones de confianza de OpenID Connect (OIDC) administran las solicitudes y las cargas útiles de respuesta desde este punto de conexión. Los tokens proporcionan una prueba de autenticación verificable, información de perfil y un mecanismo de acceso a los sistemas de backend.

El punto final del token devuelve un encabezado de `Access-Control-Allow-Origin: *` respuesta. Puede llamar al punto final del token de origen cruzado desde una aplicación basada en un navegador. Por ejemplo, puede completar una concesión de código de autorización con Proof Key for Code Exchange (PKCE) desde un cliente público. Amazon Cognito no admite políticas de

origen personalizadas de uso compartido de recursos entre orígenes (CORS) en este punto de conexión. Para obtener más información, consulte la sección de políticas de CORS de. Inicio de sesión administrado de grupos de usuarios

El servidor de autorización de OAuth 2.0 de su grupo de usuarios emite tokens web JSON (JWT) desde el punto de conexión de tokens a los siguientes tipos de sesiones:

1. Usuarios que han completado una solicitud de concesión de código de autorización. Al canjear correctamente un código, se obtienen tokens de ID, acceso y actualización.
2. Machine-to-machine Sesiones (M2M) 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 de token de actualización devuelve tokens de acceso e ID nuevos.

> **Note**
>
> Los usuarios que inicien sesión con una concesión de código de autorización en el inicio de sesión administrado o mediante federación siempre pueden actualizar los tokens desde el punto de conexión del token. Los usuarios que inician sesión con las operaciones de API `InitiateAuth` y `AdminInitiateAuth` pueden actualizar los tokens con el punto de conexión del token cuando los dispositivos recordados no estén activos en el grupo de usuarios. Si la funcionalidad de dispositivos recordados está activa, actualice los tokens con la operación de actualización de token de la API o el SDK que corresponda para su cliente de aplicación.

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 proteger la aplicación, utilice la PKCE en los eventos de inicio de sesión con 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 la PKCE, consulte IETF RFC 7636.

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 Application-specific configuración con clientes de aplicaciones. Puede obtener más información sobre la autorización de M2M, las concesiones de credenciales de cliente y la autorización con ámbitos de token de acceso en Ámbitos, M2M y servidores de recursos.

Para recuperar información sobre un usuario a partir de su token de acceso, páselo a su El punto de conexión userInfo o a una solicitud de API GetUser. El token de acceso debe contener los ámbitos adecuados para estas solicitudes.

En qué se diferencia Amazon Cognito de la especificación OIDC

Amazon Cognito no implementa el modelo de `offline_access` alcance de OpenID Connect para controlar la emisión de tokens de actualización. En su lugar, el punto final del token devuelve un token de actualización siempre que lo recibe`authorization_code`, y usted controla la `grant_type` vida útil del token de actualización mediante la configuración de validez del token de actualización del cliente de la aplicación, en lugar de hacerlo mediante una solicitud. `offline_access` Solicitar el `offline_access` ámbito no tiene ningún efecto y, al omitirlo, no se suprime el token de actualización.

Formateado de una solicitud POST en el punto de conexión de token

El punto de enlace `/oauth2/token` solo admite `HTTPS POST`. Este punto de conexión no es interactivo para el usuario. Gestione las solicitudes de token con una biblioteca OpenID Connect (OIDC) en su aplicación.

El punto de conexión de token admite la autenticación `client_secret_basic` y `client_secret_post`. Para obtener más información sobre la especificación OIDC, consulte Client Authentication. 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

Puede pasar los siguientes parámetros en el encabezado de su solicitud al punto de conexión del token.

`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 clienteabcdef01234567890, que utiliza la Base64-encoded versión de la cadena`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

A continuación, encontrará algunos parámetros que puede solicitar en formato `x-www-form-urlencoded` en el cuerpo de la solicitud al punto de conexión del token.

`grant_type`

Obligatorio.

El tipo de concesión OIDC que desea solicitar.

Debe ser `authorization_code`, `refresh_token` o `client_credentials`. Puede solicitar un token de acceso a un ámbito personalizado desde el punto de conexión del token en las siguientes condiciones:

- Ha habilitado el ámbito solicitado en la configuración del cliente de aplicación.
- Ha configurado el cliente de aplicación con un secreto de cliente.
- Has activado el cliente de la aplicación para la concesión de las credenciales del cliente. Puede habilitar esta concesión de una de las siguientes maneras:
  - Para la autenticación de máquina a máquina (M2M) de dominio alojado, `client_credentials` agréguela a la del cliente de la aplicación. `AllowedOAuthFlows`
  - Para la autenticación M2M sin dominio, agréguela a la del cliente de la aplicación. `ALLOW_CLIENT_TOKEN_AUTH` `ExplicitAuthFlows` Este flujo requiere un secreto de cliente y debe ser el único flujo de autenticación del cliente de la aplicación.

Si no habilitas ninguna de estas opciones, el punto final del token devuelve un error de clase no autorizado.

> **Note**
>
> El punto de conexión del token devuelve un token actualizado solo cuando el `grant_type` es `authorization_code`.

`client_id`

Opcional. No es obligatorio cuando proporciona el ID de cliente de aplicación en el encabezado *`Authorization`*.

El ID de un cliente de aplicaciones de su grupo de usuarios. Especifique el mismo cliente de aplicación que ha autenticado a su usuario.

Debe proporcionar este parámetro si el cliente es público y no tiene un secreto o tiene `client_secret` en la autorización `client_secret_post`.

`client_secret`

Opcional. No es obligatorio cuando proporciona el secreto del cliente en el encabezado de la *`Authorization`* y cuando el cliente de aplicación no tiene ningún secreto.

El secreto del cliente de aplicación, si este lo tiene, para la autorización `client_secret_post`.

`scope`

Opcional.

Puede ser una combinación de cualquier ámbito asociado con el cliente de su aplicación. Si la solicitud incluye un ámbito definido en un servidor de recursos, Amazon Cognito comprueba si ese ámbito está permitido para el cliente de la aplicación solicitado. Si el ámbito no está permitido, Amazon Cognito rechaza la solicitud completa en lugar de emitir un token para los ámbitos restantes. Las solicitudes que combinan los ámbitos permitidos y no permitidos fallan por el mismo motivo.

Amazon Cognito rechaza un osciloscopio inutilizable en lugar de descartarlo y devolver un token para el resto. Se devuelve una solicitud rechazada `HTTP 400` con un error en el cuerpo de la respuesta. El valor del error depende del motivo por el que no se pudo utilizar el ámbito. Cuando existe un ámbito pero no está permitido para el cliente de la aplicación, Amazon Cognito vuelve. `invalid_grant` Cuando un ámbito hace referencia a un servidor de recursos o a un ámbito personalizado que no existe, Amazon Cognito regresa. `invalid_scope` Si no proporciona este parámetro de solicitud, el servidor de autorización devuelve una reclamación `scope` de token de

acceso con todos los ámbitos de autorización que haya habilitado en la configuración del cliente de aplicación. Puede solicitar cualquiera de los ámbitos permitidos para el cliente de aplicación solicitado: los ámbitos estándar, los ámbitos personalizados de los servidores de recursos y el ámbito de autoservicio del usuario `aws.cognito.signin.user.admin`.

`redirect_uri`

Opcional. No es obligatorio para la concesión de credenciales de clientes.

Debe ser la misma `redirect_uri` que el que se utilizó para obtener `authorization_code` en `/oauth2/authorize`.

Debe proporcionar este parámetro si `grant_type` es `authorization_code`.

`refresh_token`

Opcional. Se usa solo cuando el usuario ya tiene un token de actualización y necesita nuevos identificadores y tokens de acceso.

Para generar nuevos tokens de acceso e identificación para la sesión de un usuario, establezca el valor de `refresh_token` en un token de actualización válido emitido por el cliente de aplicación solicitado.

Devuelve un nuevo token de actualización con un nuevo ID y un nuevo token de acceso cuando la rotación del token de actualización está activa; de lo contrario, solo devuelve los tokens de ID y acceso. Si el token de acceso original estaba vinculado a un recurso de API, el nuevo token de acceso mantiene la URL de la API solicitada en la reclamación `aud`.

`code`

Opcional. Solo es obligatorio en las concesiones de códigos de autorización.

El código de autorización de una concesión de código de autorización. Debe proporcionar este parámetro si la solicitud de autorización incluye un `grant_type` de `authorization_code`.

`aws_client_metadata`

Opcional.

Información que desea transferir a Desencadenador de Lambda anterior a la generación del token en los flujos de autorización máquina a máquina (M2M). Su aplicación puede recopilar información contextual sobre la sesión y pasarla a este parámetro. Cuando pasa `aws_client_metadata` en formato URL-encoded JSON, Amazon Cognito lo incluye en el evento de entrada de la función Lambda de activación. La versión previa al evento de activación

del token o la versión de activación global de Lambda deben estar configuradas para la versión tres o posterior. Si bien Amazon Cognito acepta solicitudes a este punto de conexión en los flujos M2M de códigos de autorización y credenciales de cliente, su grupo de usuarios solo pasa `aws_client_metadata` al desencadenador Antes de la generación del token desde las solicitudes de credenciales del cliente.

`code_verifier`

Opcional. Solo es obligatorio si proporcionó los parámetros `code_challenge_method` y `code_challenge` en su solicitud de autorización inicial.

El verificador de código generado a partir del cual su solicitud calculó el `code_challenge` en una solicitud de concesión de código de autorización con PKCE.

Intercambio de un código de autorización para los tokens

La siguiente solicitud genera tokens de ID, acceso y actualización correctamente después de la autenticación con una concesión de código de autorización. La solicitud transmite el secreto del cliente en formato `client_secret_basic` en el encabezado de `Authorization`.

```
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
```

La respuesta envía nuevos tokens de ID, acceso y actualización para el usuario, con metadatos adicionales.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
    "id_token": "eyJra2example",
    "refresh_token": "eyJj3example",
    "token_type": "Bearer",
    "expires_in": 3600
```

```
}
```

Credenciales de cliente con autorización básica

La siguiente solicitud de una aplicación M2M solicita la concesión de credenciales de cliente. Como las credenciales de cliente requieren un secreto de cliente, la solicitud se autoriza con un encabezado de `Authorization` derivado del ID y el secreto del cliente de aplicación. La solicitud da como resultado un token de acceso con los dos ámbitos solicitados. La solicitud también incluye metadatos del cliente que proporcionan IP-address información y un token emitido al usuario en nombre del que se concede la concesión. Amazon Cognito transfiere los metadatos del cliente al desencadenador de Lambda Antes de la generación del token.

```
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%2Fscope1%20resourceServerIdentifier2%2Fscope2&
&aws_client_metadata=%7B%22onBehalfOfToken%22%3A%22eyJra789ghiEXAMPLE%22,
%20%22ClientIpAddress%22%3A%22192.0.2.252%22%7D
```

Amazon Cognito transfiere los metadatos del cliente al desencadenador de Lambda Antes de la generación del token.

```
{
    version: '3',
    triggerSource: 'TokenGeneration_ClientCredentials',
    region: 'us-east-1',
    userPoolId: 'us-east-1_EXAMPLE',
    userName: 'ClientCredentials',
    callerContext: {
        awsSdkVersion: 'aws-sdk-unknown-unknown',
        clientId: '1example23456789'
    },
    request: {
        userAttributes: {},
        groupConfiguration: null,
        scopes: [
           'resourceServerIdentifier1/scope1',
           'resourceServerIdentifier2/scope2'
```

```
        ],
        clientMetadata: {
            'onBehalfOfToken': 'eyJra789ghiEXAMPLE',
            'ClientIpAddress': '192.0.2.252'
        }
    },
    response: { claimsAndScopeOverrideDetails: null }
}
```

La respuesta devuelve un token de acceso. Las concesiones de credenciales de cliente son para la autorización máquina a máquina (M2M) y solo devuelven tokens de acceso.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "access_token": "eyJra1example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Credenciales de cliente con autorización en el cuerpo POST

La siguiente solicitud de concesión de credenciales de cliente incluye el parámetro `client_secret` en el cuerpo de la solicitud y no incluye un encabezado de `Authorization`. Esta solicitud usa la sintaxis de autorización `client_secret_post`. La solicitud da como resultado un token de acceso con el ámbito solicitado. La solicitud también incluye metadatos del cliente que proporcionan IP-address información y un token emitido al usuario en nombre de quien se concede la concesión. Amazon Cognito transfiere los metadatos del cliente al desencadenador de Lambda Antes de la generación del token.

```
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%2Fmy_custom_scope&
client_secret=9example87654321&
aws_client_metadata=%7B%22onBehalfOfToken%22%3A%22eyJra789ghiEXAMPLE%22,
%20%22ClientIpAddress%22%3A%22192.0.2.252%22%7D
```

Amazon Cognito transfiere los metadatos del cliente al desencadenador de Lambda Antes de la generación del token.

```
{
    version: '3',
    triggerSource: 'TokenGeneration_ClientCredentials',
    region: 'us-east-1',
    userPoolId: 'us-east-1_EXAMPLE',
    userName: 'ClientCredentials',
    callerContext: {
        awsSdkVersion: 'aws-sdk-unknown-unknown',
        clientId: '1example23456789'
    },
    request: {
        userAttributes: {},
        groupConfiguration: null,
        scopes: [
           'resourceServerIdentifier1/my_custom_scope'
        ],
        clientMetadata: {
            'onBehalfOfToken': 'eyJra789ghiEXAMPLE',
            'ClientIpAddress': '192.0.2.252'
        }
    },
    response: { claimsAndScopeOverrideDetails: null }
}
```

La respuesta devuelve un token de acceso. Las concesiones de credenciales de cliente son para la autorización máquina a máquina (M2M) y solo devuelven tokens de acceso.

```
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"
}
```

Concesión de código de autorización con PKCE

El siguiente ejemplo de solicitud realiza una solicitud de autorización que incluye los parámetros `code_challenge_method` y `code_challenge` en una solicitud de concesión de código de autorización con PKCE.

```
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
```

La respuesta devuelve tokens de ID, acceso y actualización desde la verificación correcta de PKCE realizada por la aplicación.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
    "id_token": "eyJra2example",
    "refresh_token": "eyJj3example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Actualización de tokens sin rotación de tokens de actualización

El siguiente ejemplo de solicitud proporciona un token de actualización a un cliente de aplicación en el que la rotación del token de actualización está inactiva. Como el cliente de aplicación tiene un secreto de cliente, la solicitud proporciona un encabezado de `Authorization`.

```
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
```

La respuesta devuelve nuevos tokens de acceso e ID.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
    "id_token": "eyJra2example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Actualización de tokens con rotación de tokens de actualización

El siguiente ejemplo de solicitud proporciona un token de actualización a un cliente de aplicación en el que la rotación del token de actualización está activa. Como el cliente de aplicación tiene un secreto de cliente, la solicitud proporciona un encabezado de `Authorization`.

```
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
```

La respuesta devuelve nuevos tokens de acceso, ID y actualización.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "access_token": "eyJra1example",
```

```
    "id_token": "eyJra2example",
    "refresh_token": "eyJj4example",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Ejemplos de respuestas negativas

Las solicitudes con un formato incorrecto generan errores en el punto de conexión del token. A continuación, encontrará un mapa general del cuerpo de la respuesta cuando las solicitudes de token generan un 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.

Una solicitud de token con un valor `redirect_uri` que no coincide con el valor de la solicitud de autorización se devuelve `unauthorized_client` con un `error_description` de`invalid_redirect`.

`unsupported_grant_type`

Se devuelve si `grant_type` es distinto de `authorization_code`, `refresh_token` o `client_credentials`.

El punto de conexión de los atributos de usuario

Cuando OIDC emite tokens de ID que contienen atributos de usuario, OAuth 2.0 implementa el punto de conexión `/oauth2/userInfo`. Un usuario o cliente autenticado recibe un token de acceso con una notificación `scopes`. Esta notificación determina los atributos que debe devolver el servidor de autorización. Cuando una aplicación presenta un token de acceso al punto de conexión `userInfo`, el servidor de autorización devuelve en el cuerpo de la respuesta los atributos del usuario que entran en los límites establecidos por los ámbitos del token de acceso. La aplicación puede recuperar información sobre un usuario desde el punto de conexión `userInfo` siempre que disponga de un token de acceso válido que tenga al menos una notificación de ámbito `openid`.

El punto de conexión `userInfo` es un punto de conexión userInfo de OpenID Connect (OIDC). Responde con los 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 su usuario se autentica con ese IdP, Amazon Cognito intercambia silenciosamente un código de autorización con el punto de conexión `token` del IdP. Su grupo de usuarios pasa el

token de acceso del IdP para autorizar la recuperación de la información del usuario desde el punto de conexión `userInfo` del IdP.

Los ámbitos en el token de acceso de un usuario vienen determinados por el parámetro de solicitud `scopes` en las solicitudes de autenticación o por los ámbitos que añade el desencadenador de Lambda Antes de la generación del token. Puede decodificar los tokens de acceso y examinar las reclamaciones `scope` para ver los ámbitos de control de acceso que contienen. A continuación, encontrará algunas combinaciones de ámbitos que influyen en los datos devueltos desde el punto de conexión `userInfo`. El ámbito reservado `aws.cognito.signin.user.admin` de Amazon Cognito no afecta a los datos devueltos desde este punto de conexión.

Ejemplos de alcances en el token de acceso y su efecto en la respuesta de **`UserInfo`**

`openid`

Devuelve una respuesta con todos los atributos de usuario que el cliente de la aplicación puede leer.

`openid profile`

Devuelve los atributos de usuario `name`, `family_name`, `given_name`, `middle_name`, `nickname`, `preferred_username`, `profile`, `picture`, `website`, `gender`, `birthdate`, `zoneinfo`, `locale` y `updated_at`. También devuelve atributos personalizados. En los clientes de aplicación que no tienen acceso de lectura a cada atributo, la respuesta a este ámbito son todos los atributos de la especificación a los que el cliente de aplicación sí tiene acceso de lectura.

`openid email`

Devuelve la información de perfil básica y los atributos `email` y `email_verified`.

`openid phone`

Devuelve la información de perfil básica y los atributos `phone_number` y `phone_number_verified`.

OBTENGA/oauth2/userInfo

Su aplicación genera solicitudes a este punto de conexión directamente, sin pasar por 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: Respuestas negativas

Parámetros de la solicitud en el encabezado

`Authorization: Bearer <access_token>`

Pasa el token de acceso en el campo de 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: Respuestas negativas

Ejemplo: Respuesta incorrecta

```
HTTP/1.1 400 Bad Request
WWW-Authenticate: error="invalid_request",
error_description="Bad OAuth2 request at UserInfo Endpoint"
```

`invalid_request`

Falta un parámetro obligatorio en la solicitud; la solicitud incluye un valor de parámetro no admitido o tiene un formato incorrecto.

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, se ha revocado, no tiene el formato correcto o no es válido.

El punto de conexión de revocación del token

Los usuarios que tienen un token de actualización en su sesión tienen algo parecido a una cookie del navegador. Pueden renovar la sesión en curso siempre que el token de actualización sea válido. En lugar de pedirle al usuario que inicie sesión cuando su token de ID o de acceso caduquen, la aplicación puede usar el token de actualización para obtener tokens nuevos que sean válidos. Sin embargo, puede darse el caso de que externamente decida que la sesión de un usuario debe finalizar, o bien el usuario puede optar por olvidar la sesión en curso. En esa situación, puede revocar el token de actualización para que el usuario no pueda continuar con la sesión.

El punto de conexión `/oauth2/revoke` revoca un token de acceso de usuario que Amazon Cognito ha emitido inicialmente con el token de actualización proporcionado. El punto de conexión también revoca el token de actualización y 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.

PUBLICAR/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 aplicación tiene un secreto de cliente, la aplicación debe pasar el `client_id` y el `client_secret` del encabezado de la autorización mediante una autorización de HTTP básico. 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 aplicación del token que quiere revocar.

Obligatorio si el cliente es público y no tiene ningún secreto.

Ejemplo de solicitud de revocación

Esta solicitud de revocación revoca un token de actualización para un cliente de aplicación que no tiene secreto de cliente. Observe el parámetro `client_id` en el cuerpo de la solicitud.

```
POST /oauth2/revoke HTTP/1.1
Host: mydomain.auth.us-east-1.amazoncognito.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
token=2YotnFZFEjr1zCsicMWpAA&
client_id=1example23456789
```

Esta solicitud de revocación revoca un token de actualización para un cliente de aplicación que tiene un secreto de cliente. Observe el encabezado `Authorization` que contiene un ID de cliente y un secreto de cliente codificados, pero no `client_id` en el cuerpo de la solicitud.

```
POST /oauth2/revoke HTTP/1.1
Host: 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.

El punto de conexión de aserción del IdP SAML

`/saml2/idpresponse` recibe aserciones de SAML. En el inicio de sesión iniciado por el proveedor de servicios (SP-initiated), la aplicación no interactúa directamente con este punto final: el proveedor de identidad (IdP) de SAML 2.0 redirige al usuario aquí con su respuesta de SAML. Para SP-initiated iniciar sesión, configure su IdP con la ruta a `saml2/idpresponse` su URL de servicio al consumidor de aserciones (ACS). Para obtener más información acerca del inicio de sesión, consulte Inicio de sesión SAML en grupos de usuarios de Amazon Cognito.

Al IdP-initiated iniciar sesión, invoca las solicitudes a este punto final en tu aplicación después de iniciar sesión como usuario con tu proveedor de SAML 2.0. Los usuarios inician sesión con el IdP en su navegador y, a continuación, la aplicación recopila la aserción de SAML y la envía a este punto de conexión. Debe enviar las aserciones de SAML en el cuerpo de una solicitud `HTTP POST` en HTTPS. El cuerpo de la solicitud `POST` debe ser un parámetro `SAMLResponse` y un parámetro `Relaystate`. Para obtener más información, consulte Implemente el inicio de IdP-initiated sesión mediante SAML.

El punto de conexión `saml2/idpresponse` puede aceptar aserciones SAML de hasta 100 000 caracteres de longitud.

`PUBLICAR/saml2/idpresponse`

Para usar el `/saml2/idpresponse` punto final al IdP-initiated iniciar sesión, genere una solicitud POST con parámetros que proporcionen a su grupo de usuarios información sobre la sesión de su usuario.

- El cliente de aplicación en el que el usuario desee iniciar sesión.
- La URL de devolución de llamada a la que quieren que se les redirija.
- Los ámbitos de OAuth 2.0 que desea solicitar en el token de acceso de su usuario.
- El IdP que ha iniciado la solicitud de inicio de sesión.

IdP-initiated parámetros del cuerpo de la solicitud

SAMLResponse

Una afirmación de Base64-encoded SAML de un IdP asociado a un cliente de aplicación válido y a una configuración de IdP en su grupo de usuarios.

RelayState

Un parámetro `RelayState` contiene los parámetros de solicitud que, de otro modo, se pasarían al punto de conexión `oauth2/authorize`. Para obtener más información sobre estos parámetros, consulte Autorizar 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 identidades al que desea redirigir al usuario.

idp_identifier

El identificador del proveedor de identidades al que desea redirigir el usuario.

scope

Los ámbitos de OAuth 2.0 que desea que el usuario solicite al servidor de autorización.

Ejemplo de solicitudes con respuestas positivas

Ejemplo: Solicitud POST

La siguiente solicitud sirve para la concesión de un código de autorización para un usuario desde el IdP `MySAMLIdP` en el cliente de aplicación `1example23456789`. El usuario se redirige a `https://www.example.com` con el código de autorización, que se puede intercambiar por tokens que incluyan un token de acceso con los ámbitos de OAuth 2.0 `openid`, `email` y `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

A continuación se muestra una 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 Ejemplo: concesión de código de autorización. Amazon Cognito admite la clave de prueba para el intercambio de código (PKCE) en las concesiones de códigos 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 una autenticación realizada a través de la API de grupos de usuarios de Amazon Cognito, pero la API no emite tokens de acceso con ámbitos que no sean `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 `Ejemplo: concesión simbólica (implícita) sin el ámbito de 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 de cliente son una concesión solo con autorización para el acceso de máquina a máquina. 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 Ámbitos, M2M y servidores de recursos.

La concesión de credenciales de cliente añade costes a su factura. AWS Para obtener más información, consulte Precios de Amazon Cognito.

Token de actualización

Puede solicitar la concesión de un token de actualización directamente desde el Punto de conexión de token. Esta adjudicación devuelve tokens de acceso e ID nuevos a cambio de un token de actualización válido.

Para obtener más información sobre estas concesiones y su implementación, consulte How to use OAuth 2.0 in Amazon Cognito: Learn about the different OAuth 2.0 grants en el blog de AWS sobre seguridad.

Uso de la PKCE en las concesiones de códigos de autorización

Amazon Cognito admite la autenticación con clave de prueba para el intercambio de códigos (PKCE) en las concesiones de códigos de autorización. La PKCE es una extensión de la concesión de códigos de autorización de OAuth 2.0 para clientes públicos. Este tipo de autenticación evita que se canjeen códigos de autorización interceptados.

Cómo utiliza Amazon Cognito la PKCE

Para iniciar la autenticación con la 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. Pase la cadena hash al Autorizar punto de conexión como parámetro `code_challenge` en el cuerpo de la solicitud. Cuando la aplicación intercambie el código de autorización por tokens, debe incluir la cadena verificadora de código en texto simple como parámetro `code_verifier` en el cuerpo de la solicitud al Punto de conexión de token. Amazon Cognito realiza la misma operación de hash y codificación en el verificador de código. Amazon Cognito solo devuelve los tokens de ID, acceso y actualización si concluye que el verificador de código genera el mismo desafío de código que ha recibido en la solicitud de autorización.

Implementación del flujo de concesión de autorizaciones con la 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 aplicación y configure el inicio de sesión administrado durante el asistente.

   a. Si crea un nuevo grupo de usuarios, configure un cliente de aplicación y el inicio de sesión administrado durante la configuración guiada.

   b. Si configura un grupo de usuarios ya existente, añada un dominio y un cliente de aplicación público, si aún no lo ha hecho.

4. Genere una cadena alfanumérica aleatoria, normalmente un identificador único universal (UUID), para crear un desafío de código para la PKCE. Esta cadena es el valor del parámetro `code_verifier` que enviará en la solicitud al Punto de conexión de token.

5. Aplique la función hash a la cadena `code_verifier` con el algoritmo SHA256. Codifique el resultado de la operación de hash en base64. Esta cadena es el valor del parámetro `code_challenge` que enviará en la solicitud al Autorizar punto de conexión.

   El siguiente ejemplo de Python genera un `code_verifier` y calcula el `code_challenge`:

```
#!/usr/bin/env python3

import secrets
from base64 import urlsafe_b64encode
from hashlib import sha256
from string import ascii_letters
from string import digits

# use the secrets module for cryptographically strong random values
alphabet = ascii_letters + digits
code_verifier = ''.join(secrets.choice(alphabet) for _ in range(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}")
```

   A continuación se muestra un ejemplo de salida de script de Python:

```
code challenge: Eh0mg-OZv7BAyo-tdv_vYamx1boOYDulDklyXoMDtLg
code verifier: 9D-aW_iygXrgQcWJd0y0tNVMPSXSChIc2xceDhvYVdGLCBk-
JWFTmBNjvKSdOrjTTYazOFbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFGyE8yA-05-_v7Dxf3EIYJH
```

6. Complete el inicio de sesión con el inicio de sesión administrado mediante una solicitud de concesión de código de autorización con PKCE. A continuación se muestra un ejemplo de URL:

```
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://
www.example.com&code_challenge=Eh0mg-0Zv7BAyo-
tdv_vYamx1boOYDulDklyXoMDtLg&code_challenge_method=S256
```

7. Obtenga la autorización `code` y canjéela por tokens con el punto de conexión del token. A continuación, se muestra un ejemplo de solicitud:

```
POST /oauth2/token HTTP/1.1
Host: mydomain.auth.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. Revise la respuesta. Contiene los identificadores de identificación, acceso y actualización. Para obtener más información sobre el uso de tokens de grupos de usuarios de Amazon Cognito, consulte Descripción de los tokens web JSON (JWT) de los grupos de usuarios.

Respuestas de error de federación y de inicio de sesión administrado

El proceso de inicio de sesión en el inicio de sesión administrado o federado puede devolver 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 un inicio de sesión administrado.

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 tokens web JSON (JWT) de su usuario, Amazon Cognito descubre las claves web JSON (JWK) que su 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.

Implementación y operación

Una vez que el diseño de Amazon Cognito esté listo, lo utilizará como parte de un servicio de producción. En esta sección, se abordan los aspectos operativos relacionados con la ejecución de Amazon Cognito a escala: tasa de solicitudes y cuotas de recursos, capacidad aprovisionada, replicación multirregional, etiquetado de recursos y solución de problemas. Para obtener información sobre el registro y la supervisión (CloudWatch métricas e historial de CloudTrail eventos), consulte el capítulo sobre seguridad. Registro y monitoreo en Amazon Cognito

Temas

- Cuotas en Amazon Cognito
- Multi-Region replicación para grupos de usuarios
- Etiquetado de recursos de Amazon Cognito
- Resolución de problemas en Amazon Cognito

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
- Administración de las cuotas de la tasa de solicitudes de la API
- Categorías y cuotas de las tasas de solicitudes de operaciones de API de grupos de usuarios de Amazon Cognito

- Cuotas de las tasas de solicitudes de operaciones de API de grupos de identidades de Amazon Cognito (identidades federadas)
- Cuotas sobre el número y el tamaño de los recursos

Descripción de las cuotas de la tasa de solicitudes de la API

Categorización de cuotas

Amazon Cognito aplica una tasa máxima de solicitudes a las operaciones de API. Para obtener más información sobre las operaciones de API que Amazon Cognito pone a disposición del usuario, consulte las guías de referencia de las API para los grupos de usuarios y los grupos de identidades. En el caso de los grupos de usuarios, estas operaciones se agrupan por categorías de casos de uso común, como `UserAuthentication` o `UserCreation`. Para obtener una lista de operaciones de API de grupo de usuarios por categorías, 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 hacer un seguimiento del uso por categorías de grupos de usuarios y grupos de identidades. Si la tasa de solicitudes de los grupos de usuarios de Amazon Cognito supera una cuota, puede comprar más capacidad. Puede hacer un seguimiento del uso de la cuota del grupo de usuarios por categoría y adquirir aumentos de cuota en la consola de 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 una 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 categoría `UserAuthentication` incluye cuatro operaciones en la API de grupos de usuarios de Amazon Cognito: `AdminInitiateAuth`, `InitiateAuth`, `AdminRespondToAuthChallenge` 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 incluye varios desafíos de autenticación configurados en las 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 problemas de autenticación, las solicitudes adicionales se tendrán en cuenta para la cuota de `UserAuthentication` categorías. Multi-factor la autenticación (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 la cuota para la categoría `UserAuthentication` es de 80 RPS, puede llamar a `RespondToAuthChallenge` o `AdminRespondToAuthChallenge` hasta 240 RPS (3 * 80 RPS). Si el grupo de usuarios pide cuatro rondas de desafíos por autenticación y 70 usuarios inician sesión por segundo, la cantidad total de `RespondToAuthChallenge` será 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 la categoría, que está configurada en 80 RPS por 30 PS, Amazon Cognito limita las solicitudes de la 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. Al vincular usuarios federados con usuarios locales, con la federación de SAML u OIDC, el usuario local cuenta como una MAU del directorio empresarial o`EnterpriseMAU`, independientemente de si el usuario inicia sesión directamente o mediante una federación. Para obtener más información, consulte Precios de Amazon Cognito.

- Sign-up o creación administrativa de un usuario. La importación de archivos CSV de usuario no se tiene en cuenta en el recuento de MAU.
- Confirmación de la cuenta de usuario o verificación de atributos.
- Sign-in y respuesta al desafío. Las operaciones que autorice con el token de acceso de usuario que haya iniciado sesión actualmente no contribuyen al recuento de MAU; sin embargo, dado que al iniciar sesión se generan tokens de acceso, estas operaciones indican que el usuario asociado es un MAU.
- Sign-out y revocación simbólica.
- Autoservicio de restablecimiento de contraseñas y configuración de las contraseñas de los usuarios como administrador. Restablecer las contraseñas de los usuarios como administrador (AdminResetUserPassword) no contribuye al recuento de MAU.
- Cambios en los atributos de usuario o pertenencia a grupos.
- Consulta de los atributos detallados de un usuario como administrador.

> **Note**
>
> La categoría Consulta los atributos detallados de un usuario como administrador incluye la operación de la API AdminGetUser, pero no. ListUsers Una consulta detallada, usuario por usuario, en un grupo grande de usuarios puede tener un impacto significativo en su AWS factura. Para evitar un costo adicional, recopile los datos de los usuarios con `ListUsers` o almacene la información de los usuarios en una base de datos externa.

No se le cobrará por las sesiones adicionales de ningún usuario activo ni por los usuarios que no estuvieron activos durante un mes natural. En un mes en el que haya cambiado el plan de características del grupo de usuarios entre las opciones Lite, Essentials y Plus, su factura de ese mes se calculará a partir de la suma de los usuarios activos mensuales (MAU) de cada nivel, y cada MAU se asigna al nivel asignado con el precio más alto cuando el usuario estaba activo. Por ejemplo:

1. Al comenzar el mes, su grupo de usuarios está inscrito en el plan de características Plus.
2. El usuario A inicia sesión el primer día del mes.
3. El usuario B inicia sesión el primer y el último día del mes.
4. El décimo día del mes, cambia su plan de características a Essentials.
5. El usuario C inicia sesión el último día del mes.

En este escenario, el usuario A y el usuario B son MAU del tipo Plus y el usuario C es un MAU del tipo Essentials.

MAU Lite

Un usuario que estaba activo al menos una vez al mes cuando el grupo de usuarios estaba en el plan de características Lite y que nunca estuvo activo cuando el grupo de usuarios estaba en los planes Essentials o Plus.

MAU Essentials

Un usuario que estaba activo al menos una vez al mes cuando el grupo de usuarios estaba en el plan de características Essentials y que nunca estuvo activo cuando el grupo de usuarios estaba en el plan Plus.

MAU Plus

Un usuario que estaba activo al menos una vez al mes cuando el grupo de usuarios estaba en el plan Plus.

Para obtener más información, consulte Planes de características de grupo de usuarios.

Administración de las cuotas de la tasa de solicitudes de la API

Identificación de los requisitos de cuota

> **Important**
>
> Si aumenta las cuotas de Amazon Cognito para categorías como `UserAuthentication`, `UserCreation` o `AccountRecovery`, es posible que deba aumentar también las cuotas de otros 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 supone una relación pico a promedio de 3:1 al contabilizar la varianza de la frecuencia de inicio de sesión de usuarios durante el periodo de ocho horas, necesita la cuota de `UserAuthentication` de su preferencia 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.

Optimización de las tasas de solicitudes para cumplir los límites de cuota

Dado que aumentar los límites de velocidad de las API añade costes a tu AWS factura, considera la posibilidad de realizar ajustes en tu modelo de uso antes de solicitar un aumento de la cuota. A continuación se muestran algunos ejemplos de la arquitectura de aplicaciones que optimizan 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 las que puede basarse AWS.

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.

Validación de tokens web JSON (JWT) en el cliente

Las aplicaciones deben validar tokens JWT antes de confiar en ellos. Puede verificar la firma y la validez de los tokens del cliente sin tener que 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 Verificar un token web JSON.

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.

Almacenamiento en cache de 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, consulte Administración de la caducidad y el almacenamiento en caché de los tokens del grupo de usuarios. En lugar de generar solicitudes de API para consultar la información del usuario, almacene en caché los tokens de ID hasta que caduquen y lea los atributos de usuario en la memoria caché.

Para obtener más información sobre cómo trabajar con las tasas de solicitudes de API en 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 costos

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 cada uno de 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 se integran en Service Quotas, una interfaz de consola para ver y administrar 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. Cuando la cuenta haya estado activa durante un cierto período de 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 la cuota actual. La columna Utilización muestra la tasa actual de uso de la cuota. Las cuotas ajustables de solicitudes por segundo (RPS) de grupos de usuarios de Amazon Cognito muestran el 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.

Seguimiento de los usuarios activos mensuales (MAU)

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 ha tenido activos en un período de tiempo determinado. Con esta información, puede calcular cómo afecta el aumento de los usuarios activos de sus aplicaciones a las cuotas de su modelo de uso. Imagine, por ejemplo, que las aplicaciones combinadas en el oeste de EE. UU. (Oregón) dan como resultado 2 millones de usuarios activos en un mes y que su categoría `UserAuthentication` recibe errores de limitación ocasionales con la cuota predeterminada de 120 solicitudes por segundo (RPS). El mes anterior, antes del éxito de la campaña publicitaria, tenía 1 millón de MAU y las aplicaciones no superaron en ningún momento las 80 RPS. Si prevé que se producirá un aumento similar tras la emisión de un anuncio de televisión nuevo, puede adquirir, por ejemplo, 40 RPS adicionales para dar cabida al siguiente millón de usuarios con una cuota ajustada de 160 RPS.

Revisión de los MAU

Acceda a la consola de AWS Billing y consulte una factura reciente. En Cargos por servicio, puede aplicar un filtro por Cognito para ver un desglose de los MAU de 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 en cada uno. Región de AWS Puede adquirir un aumento de las cuotas ajustables de la tasa de solicitud de API de grupos de usuarios de Amazon Cognito. Para las categorías de grupos de usuarios ajustables, puede solicitar un límite máximo a nivel de cuenta superior en Service Quotas y, a continuación, utilizar la API de aprovisionamiento de Amazon Cognito para establecer la tasa deseada. Para obtener más información, consulte Administrar los límites aprovisionados.

Para solicitar un límite máximo a nivel de cuenta superior, utilice la consola Service Quotas o las operaciones `ListAWSDefaultServiceQuotas` de la API Service Quotas y. `RequestServiceQuotaIncrease`

- Para solicitar un aumento del límite máximo a nivel de cuenta mediante la consola de Service Quotas, consulte Solicitar un aumento de cuota de API en la Guía del usuario de Service Quotas.
- Algunos aumentos del límite máximo a nivel de cuenta se aprueban automáticamente. De lo contrario, tiene AWS como objetivo completar las solicitudes de aumento de cuota en un plazo de 10 días. Sin embargo, varias factores 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.

> **Note**
>
> Solo se pueden aumentar las cuotas ajustables.

> **Note**
>
> Los aumentos de cuota son configuraciones regionales. Un aumento de cuota en una de ellas Región de AWS no afectará a las cuotas en otras regiones. Si necesitas cuotas de API más altas en varias regiones, debes solicitar un aumento de cuota en cada región por separado.

> **Note**
>
> Solicitar un aumento del límite máximo a nivel de cuenta es solo el primer paso. Una vez aprobada la solicitud, debe llamar `UpdateProvisionedLimit` o utilizar la consola de Amazon Cognito para establecer el límite aprovisionado. Para obtener más información, consulte Administrar los límites aprovisionados.

Administrar los límites aprovisionados

Los grupos de usuarios de Amazon Cognito admiten un modelo de cuotas de dos niveles para cuotas de API ajustables. Con este modelo, puede aumentar o disminuir mediante programación la tasa de solicitudes aprovisionadas sin ponerse en contacto con Support. AWS Los dos niveles funcionan de la siguiente manera:

Account-level límite máximo

: La tarifa máxima que puede aprovisionar, establecida a través de Service Quotas. Para aumentar el límite máximo a nivel de cuenta, envía una solicitud de aumento de cuota en la consola Service Quotas. Algunas solicitudes se aprueban automáticamente. Para obtener más información, consulte Solicitud de aumento de cuota.

Límite aprovisionado

: La tasa, en solicitudes por segundo (RPS), que Amazon Cognito aplica actualmente a su cuenta. Puede ajustar este valor hasta el límite máximo de su cuenta con las operaciones de la API de aprovisionamiento de Amazon Cognito.

Por ejemplo, si el límite máximo a `UserAuthentication` nivel de cuenta es de 500 RPS, puede usar la operación de `UpdateProvisionedLimit` API para establecer el límite aprovisionado en

300 RPS. Más adelante, podrás reducir el límite aprovisionado al valor predeterminado y solo se te facturará el tiempo con la tarifa más alta.

> **Important**
>
> Se le facturará la capacidad aprovisionada por encima de la tarifa predeterminada, prorrateada en función del tiempo de uso real. Para conocer los precios, consulte los precios de Amazon Cognito.

Administrar los límites aprovisionados en la consola

Puede ver y gestionar los límites aprovisionados en la consola de Amazon Cognito. Abra la página Grupos de usuarios y elija la pestaña Límites aprovisionados. En esta pestaña, puedes ver los límites aprovisionados actuales y los límites máximos a nivel de cuenta para cada categoría de API ajustable. Puedes aumentar o disminuir la tarifa aprovisionada dentro del límite máximo aprobado a nivel de cuenta.

Aprovisionamiento de operaciones de API

Amazon Cognito proporciona las siguientes operaciones de API para gestionar los límites aprovisionados. Estas operaciones requieren la autenticación y los permisos de IAM correspondientes.

GetProvisionedLimit

: Devuelve el valor límite aprovisionado actual para una categoría de API específica. La respuesta incluye el valor límite aprovisionado y el valor límite predeterminado (gratuito).

UpdateProvisionedLimit

: Establece el valor límite aprovisionado para una categoría de API específica. El valor solicitado debe estar entre el límite predeterminado y el límite máximo a nivel de cuenta de Service Quotas. Esta operación se realiza de forma inmediata.

Ambas operaciones utilizan un `LimitDefinition` parámetro que identifica el límite a gestionar. En el caso de los límites de frecuencia de las API para grupos de usuarios, la definición del límite incluye una clase `API_CATEGORY` y un mapa de atributos con la `Category` clave. Por ejemplo, para gestionar el `UserAuthentication` límite, especifique una definición de límite con`{"Category": "UserAuthentication"}`.

Example Ejemplo de GetProvisionedLimit solicitud y respuesta

El siguiente ejemplo devuelve el límite aprovisionado para la categoría de `UserAuthentication` API.

Solicitud

```
{
    "LimitDefinition": {
        "LimitClass": "API_CATEGORY",
        "Attributes": {"Category": "UserAuthentication"}
    }
}
```

Respuesta

```
{
    "Limit": {
        "LimitDefinition": {
            "LimitClass": "API_CATEGORY",
            "Attributes": {"Category": "UserAuthentication"}
        },
        "ProvisionedLimitValue": 120,
        "FreeLimitValue": 120
    }
}
```

Example Ejemplo de UpdateProvisionedLimit solicitud y respuesta

En el siguiente ejemplo, se establece el límite aprovisionado para la categoría de `UserAuthentication` API en 300 RPS.

Solicitud

```
{
    "LimitDefinition": {
        "LimitClass": "API_CATEGORY",
        "Attributes": {"Category": "UserAuthentication"}
    },
    "RequestedLimitValue": 300
}
```

Respuesta

```
{
    "Limit": {
        "LimitDefinition": {
            "LimitClass": "API_CATEGORY",
            "Attributes": {"Category": "UserAuthentication"}
        },
        "ProvisionedLimitValue": 300,
        "FreeLimitValue": 120
    }
}
```

Las siguientes acciones de IAM controlan el acceso a estas operaciones:

- `cognito-idp:GetProvisionedLimit`
- `cognito-idp:UpdateProvisionedLimit`

> **Note**
>
> Cross-account no se admite el acceso. Solo la cuenta propietaria puede gestionar los límites aprovisionados.

flujo de trabajo de aprovisionamiento

El siguiente flujo de trabajo describe cómo aumentar el límite de aprovisionamiento.

1. Determine la tarifa requerida. Para obtener más información, consulte Identificación de los requisitos de cuota.
2. Si la tarifa requerida supera el límite máximo actual a nivel de cuenta de Service Quotas, solicite un aumento en la consola de Service Quotas.
3. Una vez que se apruebe el límite máximo a nivel de cuenta, llame `UpdateProvisionedLimit` con la tarifa que desee.
4. Para reducir la tarifa aprovisionada y evitar la facturación adicional, llama `UpdateProvisionedLimit` con el valor límite predeterminado o con un valor inferior.

Consideraciones

- Los límites aprovisionados son recursos a nivel de cuenta. Se aplican a la tasa agregada de todas las solicitudes de todos los grupos de usuarios en un solo Región de AWS grupo. Cuenta de AWS
- La `LimitManagement` categoría, que incluye `GetProvisionedLimit` y`UpdateProvisionedLimit`, tiene una tarifa limitada a 1 solicitud por segundo por cuenta.
- Solo las categorías de cuotas ajustables admiten el aprovisionamiento. Para obtener una lista de categorías ajustables, consulteCategorías y cuotas de las tasas de solicitudes de operaciones de API de grupos de usuarios de Amazon Cognito.
- Si tu grupo de usuarios usa el inicio de sesión administrado, no puedes usar la API de aprovisionamiento para ajustar las `UserFederation` categorías `UserAuthentication` o. Para aumentar estas cuotas, envíe una solicitud de aumento de las cuotas de Service Quotas a través de la consola de Service Quotas o utilice el formulario de aumento del límite de servicio.
- Los cambios en las cuotas aprovisionadas son regionales. Un cambio de cuota aprovisionada en una de ellas Región de AWS no afectará a las cuotas de otras regiones.
- Se registran todas las llamadas a las operaciones de la API de aprovisionamiento. AWS CloudTrail

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) se aplica adicionalmente a todos los grupos de usuarios de una 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, sopese la

posibilidad de optimizar la aplicación. Para obtener más información, consulte Optimización de las tasas de solicitudes para cumplir 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` , `InitiateAuth` , `RespondToAuthChallenge`	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	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
`UserAuthentication` • InitiateAuth • Actualización de token con `InitiateAuth`	Operaciones con las que se autentifica a un usuario (inicia sesión). Estas operaciones están sujetas a Operaciones de API	120	Sí

Categoría	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
o el punto de conexión de token • RespondToAuthChallenge[1] • AdminInitiateAuth • AdminRespondToAuthChallenge[1] • GetTokensFromRefreshToken • Inicio de sesión administrado o inicio de sesión con interfaz de usuario alojada clásica y MFA para usuarios locales en concesiones implícitas o con código de autorización [2]	de grupo de usuarios 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	Description (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	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
`UserAccountRecovery` • ChangePassword • ConfirmForgotPassword • ForgotPassword • AdminResetUserPassword • AdminSetUserPassword • RespondToAuthChallenge[1] • AdminRespondToAuthChallenge[1] • Inicio de sesión administrado y restablecimiento de contraseñas	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 • AdminGetUserAuthFactors • GetUser	Operaciones con las que se recupera un usuario de los grupos de usuarios	120	Sí

Categoría	Description (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	Description (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	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
`UserResourceUpdate` • AdminForgetDevice • AdminUpdateAuthEventFeedback • AdminSetUserMFAPreference • AdminDeleteSoftwareToken • AdminSetUserSettings • AdminUpdateDeviceStatus • UpdateDeviceStatus • UpdateAuthEventFeedback • ConfirmDevice • SetUserMFAPreference • 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	Description (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	Description (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	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
• DescribeManagedLoginBranding • DescribeManagedLoginBrandingByClient • ListUserPoolReplicas			

Categoría	Description (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	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
• UpdateUserPoolDomain • SetRiskConfiguration • SetUICustomization • TagResource • UntagResource • CreateManagedLoginBranding • UpdateManagedLoginBranding • DeleteManagedLoginBranding • CreateUserPoolReplica • UpdateUserPoolReplica • DeleteUserPoolReplica			
`UserPoolClientRead` • DescribeUserPoolClient • ListUserPoolClients	Operaciones con las que se recupera información sobre los clientes del grupo de usuarios.[3]	15	No

Categoría	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
`UserPoolClientUpdate` • CreateUserPoolClient • DeleteUserPoolClient • UpdateUserPoolClient	Operaciones con las que se crean, actualizan y eliminan los clientes de un grupo de usuarios.[3]	15	No
`ClientAuthentication` • `client_credentials` conceda solicitudes de tipo al punto final del token • GetClientToken	Operaciones que generan credenciales que se utilizarán para autorizar solicitudes de máquina a máquina	150	No
`LimitManagement` • GetProvisionedLimit • UpdateProvisionedLimit	Operaciones que recuperan o actualizan los límites de velocidad de la API aprovisionados	1	No

[1] Una respuesta `RespondToAuthChallenge` o `AdminRespondToAuthChallenge` con un `ChallengeName` de `NEW_PASSWORD_REQUIRED` cuenta para la categoría `UserAccountRecovery`. Todas las demás respuestas al desafío cuentan para la categoría `UserAuthentication`.

[2] Cada operación de interfaz de usuario alojada durante el inicio de sesión contribuye una solicitud a la cuota. Por ejemplo, un usuario que inicia sesión y proporciona un código MFA aporta dos

solicitudes. El canje de tokens en las concesiones con códigos de autorización está sujeto a una asignación de cuota adicional equivalente a la cuota de la categoría `UserAuthentication`.

[3] Cualquier operación individual de esta categoría tiene una limitación que impide que se llame a la operación en tasas superiores a 5 RPS para un solo grupo de usuarios.

Límites de tasa de solicitudes masivas para dominios de grupos de usuarios

Las siguientes cuotas se aplican al volumen total de solicitudes a un dominio de grupo de usuarios.

Operación	Description (Descripción)	Cuota predeterminada (RPS)	Ajustable
Solicitudes desde la IP de origen	Volumen de solicitudes desde una dirección IP a un dominio	300	No
Solicitudes al cliente de aplicaciones	Volumen de solicitudes de un ID de cliente de aplicación en un dominio	300	No
Solicitudes de dominio	Volumen total de solicitudes para los servicios de un dominio de grupo de usuarios	500	No
Solicitudes de documentos clave web JSON	Volumen de solicitudes de una `jwks.json` Cuenta de AWS en una Región de AWS	50 000	No

Cuotas de las tasas de solicitudes de operaciones de API de grupos de identidades de Amazon Cognito (identidades federadas)

Operación	Description (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 flujo de trabajo mejorado.	200	Sí	Contáctese con el equipo de cuentas.
`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.

Operación	Description (Descripción)	Cuota predeterminada (RPS)[1]	Ajustable	Idoneidad para el aumento de cuota
`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.
`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

Las siguientes cuotas describen el número máximo o la longitud de los elementos que se pueden crear en los grupos de usuarios.

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

Recurso	Cuota	Ajustable	Cuota máxima
Servidores de recursos por grupo de usuarios	100	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.
Estilos de marca de inicio de sesión administrado por grupo de usuarios	20	No	N/A
Documentos de condiciones de inicio de sesión gestionados por grupo de usuarios	40	No	N/A
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
Mensajes de MFA por correo electrónico enviados a una dirección de correo electrónico cada hora por dirección IP del solicitante	5-20	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

Recurso	Cuota	Ajustable	Cuota máxima
Caracteres en el nombre del proveedor de identidad	32	No	N/A
Caracteres de una respuesta de SAML	100 000	No	N/A
Identificadores por proveedor de identidad	50	No	N/A
Identidades vinculadas a un usuario	5	No	N/A
Passkey/WebAuthn autenticadores por usuario	20	No	N/A
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 región	4	No	N/A

Recurso	Cuota	Ajustable	Cuota máxima
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 notificaciones existentes y añadidas más los ámbitos de los tokens de acceso y de identidad de una transacción 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. Esta restricción se restablece todos los días a las 09:00 UTC. 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

Las siguientes cuotas describen la configuración disponible durante la duración de los artefactos de autenticación y las sesiones de usuario en los grupos de usuarios.

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)

Los siguientes cupos describen los períodos de tiempo disponibles relacionados con los códigos para el inicio de sesión, el registro y el restablecimiento de la contraseña.

Recurso	Cuota
Sign-up período de validez del código de confirmación	24 horas
Periodo de validez del código de verificación de atributo de usuario	24 horas
Multi-factor período de validez del código de autenticación (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
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

Las siguientes cuotas describen los recursos y los límites disponibles para los trabajos de importación de usuarios.

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)

Las siguientes cuotas describen el número máximo o la longitud de los elementos que se pueden crear en los grupos de identidades.

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
Role-based reglas de control de acceso (RBAC)	25	No	N/A

Cuotas de recursos de Amazon Cognito Sync

Las siguientes cuotas describen el número máximo o la longitud de los elementos que se pueden crear en 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

Multi-Region replicación para grupos de usuarios

Con la replicación multirregional (MRR), puede crear una réplica de un grupo de usuarios adicional Región de AWS para proporcionar capacidades de continuidad empresarial y recuperación ante desastres para su infraestructura de autenticación. Con el MRR, los usuarios registrados pueden seguir autenticándose con sus aplicaciones incluso si pierde la conectividad con los recursos de una región, lo que garantiza que sus aplicaciones permanezcan disponibles.

Al configurar MRR, Amazon Cognito crea grupos de usuarios independientes con un ID de grupo de usuarios compartido. Cada grupo de usuarios de réplica aloja los servicios de autenticación para un directorio de usuarios compartido. El grupo de usuarios principal sirve como fuente autorizada para la configuración administrativa y las operaciones de escritura, como el restablecimiento de

contraseñas y el registro de usuarios. Los grupos de usuarios secundarios no pueden crear usuarios. Heredan la mayoría de los ajustes del grupo de usuarios principal y, en un estado de conmutación por error, pueden gestionar operaciones de autenticación, como el inicio de sesión de los usuarios y la generación de tokens.

Lo que debe saber sobre la replicación multirregional

- Multi-Region la replicación tiene costos adicionales independientes y requiere que su grupo de usuarios esté inscrito en el plan de funciones Essentials o Plus. No puedes habilitar el MRR en grupos de usuarios con el plan de funciones Lite.
- Debe configurar su grupo de usuarios con una clave gestionada por el cliente para varias regiones AWS KMS antes de habilitar la replicación. La clave debe estar disponible en todos los Regiones de AWS que tengan réplicas de grupos de usuarios. Para obtener más información, consulte the section called “Cifrado de datos”.
- Para que la validación de los tokens sea uniforme en todas las regiones, le recomendamos que configure su grupo de usuarios con un emisor actualizado. Para obtener más información, consulte the section called “Emisor de OIDC”.
- Los nuevos grupos de usuarios secundarios comienzan en el `INACTIVE` estado. Revise y configure los ajustes regionales antes de activar el grupo de usuarios para su uso en producción.
- Las configuraciones regionales pueden diferir entre las réplicas. Puede configurar los siguientes ajustes de forma independiente en las réplicas. Todos los demás ajustes se configuran en el grupo de usuarios principal y se sincronizan automáticamente con el secundario.
  - Configuración del correo electrónico
  - Configuración del correo electrónico para las notificaciones de protección contra amenazas
  - Configuración de SMS
  - Disparadores Lambda
  - Tags
  - Configuración de exportación de registros
  - AWS WAF ACL web
- La replicación de datos entre regiones puede provocar breves retrasos. El grupo de usuarios principal sincroniza la configuración y las actualizaciones del directorio de usuarios con el secundario y, en última instancia, este proceso es coherente.

Limitaciones de la replicación multirregional

- No puede generar nuevos usuarios en grupos de usuarios secundarios, ya sea mediante el registro o mediante la creación por parte del administrador. Los usuarios federados solo pueden iniciar sesión en un grupo de usuarios secundario en estado de conmutación por error si ya han iniciado sesión en el grupo de usuarios principal.
- Los usuarios no pueden restablecer sus contraseñas ni modificar sus perfiles en los grupos de usuarios secundarios. En estado de conmutación por error, deshabilite estas operaciones en la interfaz de usuario y haga que estén disponibles después de que la comprobación de estado restablezca el acceso al grupo de usuarios principal.
- Puede tener como máximo una réplica secundaria en una región adicional por directorio de usuarios. Cualquier grupo de usuarios que cumpla los requisitos puede tener una réplica secundaria.
- El MFA TOTP no se admite en las réplicas secundarias. Los usuarios con el MFA TOTP configurado deben autenticarse cuando el grupo de usuarios de la región principal atienda las solicitudes.
- El recuento de intentos de autenticación mediante contraseña antes del bloqueo no se sincroniza en todas las regiones. Cada réplica mantiene su propio recuento de intentos de autenticación fallidos.

Configuración de la replicación multirregional

Antes de poder habilitar la replicación multirregional, asegúrese de que su grupo de usuarios cumpla los requisitos previos: el plan de funciones Essentials o Plus y la clave KMS gestionada por el cliente para varias regiones.

Consola de administración de AWS

Para configurar la replicación multirregional para un grupo de usuarios

1. Inicie sesión en la consola de Amazon Cognito.
2. Elija User pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente de la lista o cree un grupo de usuarios nuevo.
4. Elija la pestaña Settings.
5. En el menú de navegación de la izquierda, elija Multi-Regionla replicación.

6. Elija Crear un grupo de usuarios de réplica.
7. En Región, seleccione la ubicación Región de AWS en la que desea crear el grupo de usuarios de réplica.
8. Revise el resumen de la configuración y elija Crear réplica.
9. Una vez creada la réplica, revise los ajustes de configuración regional en la tabla de comparación. Configure cualquier Region-specific configuración, como la configuración del correo electrónico, la configuración de SMS o los activadores Lambda, según sea necesario para su región de réplica.
10. Para configurar una comprobación de estado de Route 53 para su dominio, vaya al menú de servicios de dominio, edite o añada un dominio personalizado y configure un ID de comprobación de estado de Route 53.
11. Cuando esté listo para usar la réplica para el tráfico de producción, cambie el estado de la réplica de Inactiva a Activa.

API

Para crear un grupo de usuarios de réplica, utilice la CreateUserPoolReplicaoperación. En el siguiente ejemplo, se crea una réplica en la `us-west-2` región para un grupo de usuarios principal en`us-east-1`.

```
{
 "UserPoolId": "us-east-1_EXAMPLE",
 "RegionName": "us-west-2",
 "UserPoolTags": {
    "Environment": "Production",
    "Application": "MyApp"
 }
}
```

La respuesta incluye la información de la réplica:

```
{
 "Replica": {
    "RegionName": "us-west-2",
    "UserPoolArn": "arn:aws:cognito-idp:us-west-2:111122223333:userpool/us-
east-1_EXAMPLE",
    "Status": "PENDING_CREATE",
    "Role": "SECONDARY"
```

```
 }
}
```

También debe configurar su dominio para la conmutación por error. Configure un chequeo de estado en Route 53 y aplíquelo a su dominio en una UpdateUserPoolDomainsolicitud:

```
{
 "CustomDomainConfig": {
    "CertificateArn": "arn:aws:acm:us-
east-1:111122223333:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
 },
 "Domain": "auth.example.com",
 "ManagedLoginVersion": 2,
 "Routing": {
    "Failover": {
       "SecondaryRegion": "us-west-2",
       "PrimaryRoute53HealthCheckId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
 },
 "UserPoolId": "us-east-1_EXAMPLE"
}
```

Para activar la réplica para su uso en producción, utilice la siguiente UpdateUserPoolReplicaoperación:

```
{
 "UserPoolId": "us-east-1_EXAMPLE",
 "RegionName": "us-west-2",
 "Status": "ACTIVE"
}
```

La respuesta confirma el estado de la réplica actualizada:

```
{
 "Replica": {
    "RegionName": "us-west-2",
    "UserPoolArn": "arn:aws:cognito-idp:us-west-2:111122223333:userpool/us-
east-1_EXAMPLE",
    "Status": "ACTIVE",
    "Role": "SECONDARY"
 }
```

```
}
```

Operaciones de API compatibles en regiones secundarias

Amazon Cognito admite un subconjunto de operaciones de API en regiones secundarias. Las operaciones disponibles dependen del estado de la réplica. Las réplicas en `INACTIVE` estado admiten un conjunto limitado de operaciones de lectura y configuración. Las réplicas en `ACTIVE` estado admiten operaciones adicionales de autenticación y administración de sesiones. Las operaciones que no aparecen aquí solo están disponibles en la región principal.

Operaciones para regiones secundarias INACTIVAS

Los grupos de usuarios de réplica en regiones secundarias en `INACTIVE` estado permiten las siguientes operaciones de la API de Amazon Cognito.

- AdminGetDevice
- AdminGetUser
- AdminListDevices
- AdminListGroupsForUser
- AdminListUserAuthEvents
- AssociateWebACL
- CreateUserPoolDomain
- DeleteUserPoolDomain
- DeleteUserPoolReplica
- DescribeIdentityProvider
- DescribeManagedLoginBranding
- DescribeManagedLoginBrandingByClient
- DescribeResourceServer
- DescribeRiskConfiguration
- DescribeTerms
- DescribeUserPool
- DescribeUserPoolClient
- DescribeUserPoolDomain

- DisassociateWebACL
- GetGroup
- GetLogDeliveryConfiguration
- GetSigningCertificate
- GetUICustomization
- GetUserPoolMfaConfig
- ListIdentityProviders
- ListGroups
- ListResourceServers
- ListTagsForResource
- ListTerms
- ListUserPoolClients
- ListUserPoolClientSecrets
- ListUserPoolReplicas
- ListUserPools
- ListUsers
- ListUsersInGroup
- SetLogDeliveryConfiguration
- SetRiskConfiguration
- SetUserPoolMfaConfig
- TagResource
- UntagResource
- UpdateUserPool
- UpdateUserPoolDomain
- UpdateUserPoolReplica

Operaciones adicionales para las regiones secundarias ACTIVAS

Las réplicas de grupos de usuarios en regiones secundarias en `ACTIVE` estado permiten todas las operaciones anteriores, además de las siguientes operaciones de autenticación y administración de sesiones.

- AdminInitiateAuth
- AdminRespondToAuthChallenge
- AdminUpdateAuthEventFeedback
- AdminUserGlobalSignOut
- GetDevice
- GetIdentityProviderByIdentifier
- GetTokensFromRefreshToken
- GetUser
- GetUserAuthFactors
- GlobalSignOut
- InitiateAuth
- ListDevices
- ListWebAuthnCredentials
- RespondToAuthChallenge
- RevokeToken
- UpdateAuthEventFeedback

La conmutación por error en grupos de usuarios de varias regiones

Con los grupos de usuarios multirregionales, puede conmutar por error el inicio de sesión gestionado, el inicio de sesión federado y las llamadas directas a la API entre dos. Regiones de AWS La conmutación por error de inicio de sesión y federación gestionada está disponible con un dominio personalizado o un dominio con prefijo (Cognito) configurado con su grupo de usuarios. No puede configurar un dominio personalizado diferente con grupos de usuarios de réplica.

> **Note**
>
> Actualmente, Amazon Cognito no admite la conmutación por error automática con una comprobación de estado de Route 53 en las regiones AWS GovCloud (EE. UU.). Para obtener información sobre el enrutamiento del tráfico a la región secundaria de AWS GovCloud (EE. UU.), consulte. the section called “La conmutación por error en AWS GovCloud Regiones (EE. UU.)”

Conmutación por error para el inicio de sesión gestionado, la federación y la autorización de máquina a máquina

La conmutación por error está disponible cuando el grupo de usuarios principal tiene un dominio personalizado o un dominio con prefijo. El dominio de tu grupo de usuarios sirve a los recursos de OAuth 2.0, incluidos los puntos de enlace de autorización y token, y gestiona las respuestas de IdP de proveedores de federaciones de terceros, incluidos OIDC, SAML y proveedores sociales.

Para habilitar la conmutación por error, configura una comprobación de estado en Route 53 y configura el campo en tu dominio. `Routing` Tú determinas qué desencadena un estado saludable o insalubre. Cuando la comprobación de estado se encuentra en mal estado, Amazon Cognito muestra páginas de inicio de sesión gestionadas y operaciones de autenticación desde el grupo de usuarios de réplica secundario. Cuando la comprobación de estado pasa a un estado correcto, Amazon Cognito comienza a redirigir el tráfico de vuelta a la réplica principal.

En el caso de los dominios personalizados, el registro de DNS puede utilizar Route 53 o cualquier proveedor de DNS de terceros. Asegúrese de tener un registro CNAME válido en su proveedor de DNS que apunte a su alias de destino, que es una CloudFront distribución. Puede encontrar el destino del alias en la página del dominio de la consola de Amazon Cognito.

Para actualizar el ID de verificación de estado en la consola

1. Inicie sesión en la consola de Amazon Cognito.
2. Elija Grupos de usuarios y, a continuación, elija su grupo de usuarios.
3. Selecciona Dominio en Marca en el menú.
4. En la sección Dominio personalizado, elige la opción de edición y selecciona Editar conmutación por error multirregional.
5. Active la opción Habilitar la conmutación por error en varias regiones.
6. Seleccione su ID de chequeo de estado de Route 53 entre los chequeos de estado disponibles.
7. Seleccione Save changes (Guardar cambios).

Conmutación por error para las API y los SDK de Amazon Cognito

Si usa las API o los SDK de Amazon Cognito, no se utiliza un dominio personalizado y su aplicación es responsable de dirigir el tráfico al punto de enlace regional del servicio Amazon Cognito para gestionar la autenticación y otras llamadas a la API.

Si solo tiene una interfaz de aplicación que utiliza un cliente público, como una aplicación de una sola página (SPA) o una aplicación móvil, su aplicación debe ser dinámica para enrutar las llamadas a la API en consecuencia. Considere la posibilidad de utilizar un back-end de aplicaciones sin servidor para determinar por qué región debe empezar la autenticación con Amazon Cognito.

Si tiene una aplicación con un back-end, aquí puede determinar la lógica para determinar con qué grupo de usuarios debe autenticarse.

Si utiliza tanto puntos de enlace de inicio de sesión gestionados como API, utilice la misma comprobación de estado de Route 53 para determinar a qué región dirige la aplicación las llamadas a la API de Amazon Cognito.

La conmutación por error en AWS GovCloud Regiones (EE. UU.)

Actualmente, Amazon Cognito no admite la conmutación por error automática con una comprobación de estado de Route 53 en las regiones AWS GovCloud (EE. UU.). No puede configurar el `Routing` campo en un dominio de grupo de usuarios y los dominios personalizados no están disponibles. Para seguir ofreciendo conexiones gestionadas, federaciones y puntos finales de OAuth 2.0 desde la región secundaria, configure un dominio con prefijo para cada grupo de réplicas de usuarios y dirija usted mismo el tráfico entre los dominios.

Complete los siguientes pasos antes de tener que realizar la conmutación por error. Antes de empezar, cree una réplica de un grupo de usuarios en la región secundaria.

Para prepararse para la conmutación por error en el AWS GovCloud Regiones (EE. UU.)

1. En la región secundaria, cree un dominio de prefijo para el grupo de usuarios de réplica con inicio de sesión administrado. En una `CreateUserPoolDomain` solicitud, `ManagedLoginVersion` 2 configúrelo o seleccione Inicio de sesión gestionado en el menú Dominio de la consola de Amazon Cognito. Los grupos de usuarios réplicas no admiten dominios con la interfaz de usuario alojada clásica. Cada dominio tiene una Region-specific URL.
2. Si su grupo de usuarios tiene proveedores de identidad federados (IdPs), añada la URL de redireccionamiento del dominio secundario a cada IdP. Para OIDC y redes sociales IdPs, agrégala `https://`*`secondary-domain`*`/oauth2/idpresponse` como URI de redireccionamiento permitida. En el caso de SAML IdPs, `https://`*`secondary-domain`*`/saml2/idpresponse` añádelo como URL del servicio de consumo de aserciones (ACS). Si omites este paso, se producirá un error en el inicio de sesión federado después de cambiar el tráfico al dominio secundario.

3. Configura tu aplicación para que envíe solicitudes a los puntos finales de autorización, token, cierre de sesión y otros puntos finales de OAuth 2.0 del dominio de la región que atiende el tráfico. Incluye los clientes de máquina a máquina (M2M) que solicitan tokens con la concesión de credenciales del cliente.
4. Antes de enviar tráfico a la región secundaria, cambie el estado del grupo de usuarios de la réplica a. `ACTIVE`

Al enrutar el tráfico al dominio secundario, tenga en cuenta lo siguiente:

- Claves de paso: las claves de acceso que los usuarios registraron en el dominio principal no funcionan en el dominio secundario, ya que el ID de la parte que confía es diferente para cada dominio. Para iniciar sesión en un inicio de sesión gestionado en el dominio secundario, los usuarios deben utilizar otro método de inicio de sesión. Esto no afecta a las aplicaciones que usan los SDK y establecen su propio dominio como ID de la parte que confía.
- Conversión por error: Amazon Cognito no redirige el tráfico a la región principal. Cuando la región principal vuelva a estar disponible, actualice la aplicación para enviar el tráfico al dominio principal.

Para obtener información sobre el enrutamiento del tráfico de API y SDK, consultethe section called “Conmutación por error para las API y los SDK de Amazon Cognito”.

Etiquetado de recursos de Amazon Cognito

Una etiqueta es una etiqueta de metadatos que usted o AWS asigna a un AWS recurso. Cada etiqueta consta de una clave y un 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 lo ayudan a hacer lo siguiente:

- Identifique y organice sus AWS recursos. Muchos AWS servicios 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.
- Realice un seguimiento de sus AWS costes. Puede activar estas etiquetas en el Administración de facturación y costos de AWS panel de control. AWS usa etiquetas de asignación de costos para categorizar sus costos y entregarle un informe mensual de asignación de costos. 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 usar la AWS Command Line Interface API de Amazon Cognito para añadir, editar o eliminar etiquetas para grupos de usuarios e 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, espacios 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, 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 el menú Configuración y busque la pestaña 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 Guardar cambios.

Para etiquetar un grupo de identidades, navegue hasta el menú Grupos de identidades y seleccione o cree un grupo de identidades. En la pestaña Propiedades del grupo de identidades, busque Etiquetas. Seleccione Agregar etiqueta.

AWS CLI ejemplos

AWS CLI Proporciona comandos que le ayudan a administrar las etiquetas que asigna a los grupos de usuarios y grupos 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 Comando **`tag-resource 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 **Comando tag-resource 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 **Comando list-tags-for-resource 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 **Comando list-tags-for-resource 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 **Comando untag-resource 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 **Comando untag-resource 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 **`Comando create-user-pool`** 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
```

Key-value los pares de etiquetas deben estar en el formato. *`key`*=*`value`* Si va a agregar varias etiquetas, especifíquelas en una lista separada por comas.

Example **`Comando create-identity-pool 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
```

Key-value los pares de etiquetas deben estar en 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

CloudFormation permisos de etiquetado

Al crear o actualizar los recursos del grupo de usuarios o del grupo de identidades de Amazon Cognito AWS CloudFormation, CloudFormation puede propagar etiquetas a nivel de pila y etiquetas a nivel de sistema a sus recursos. Para permitir la propagación de esta etiqueta, el director de IAM que implementa la pila debe tener permisos de etiquetado además de los permisos de creación de recursos.

Por ejemplo, para crear un grupo de usuarios con`cognito-idp:CreateUserPool`, la persona que llama también debe tener los siguientes permisos:

- `cognito-idp:TagResource`
- `cognito-idp:UntagResource`
- `cognito-idp:ListTagsForResource`

Del mismo modo, para crear un grupo de identidades con`cognito-identity:CreateIdentityPool`, la persona que llama también debe tener los siguientes permisos:

- `cognito-identity:TagResource`
- `cognito-identity:UntagResource`
- `cognito-identity:ListTagsForResource`

Sin estos permisos, la creación o actualización de la pila puede fallar al CloudFormation intentar aplicar etiquetas al recurso. Si aparece un error de acceso denegado relacionado con el etiquetado durante una operación de apilado, compruebe que su política de IAM incluya estos permisos.

Resolución de problemas en Amazon Cognito

En este capítulo, se proporcionan soluciones a los problemas habituales que pueden surgir al trabajar con Amazon Cognito. Las implementaciones de Amazon Cognito pueden enfrentarse a diversos desafíos en los flujos de autenticación, las configuraciones de grupos de usuarios y las configuraciones de federación de identidades. Tanto si está desarrollando una nueva aplicación como si está manteniendo una existente, esta guía de solución de problemas le ayudará a identificar y resolver rápidamente los problemas comunes.

Errores en la configuración de dominios personalizados

Al configurar nombres de dominio personalizados en Amazon Cognito, es posible que reciba mensajes de error. Los errores más comunes incluyen problemas de validación, problemas con los certificados o conflictos de dominio.

`El dominio personalizado no es un subdominio válido`

Este error indica un problema con la resolución del DNS del dominio principal. Amazon Cognito no admite dominios de nivel superior y requiere que el dominio principal tenga un registro A de DNS para su validación.

Problema

Este error indica un problema con la resolución del DNS del dominio principal. Amazon Cognito no admite dominios de nivel superior y requiere que el dominio principal tenga un registro A de DNS para su validación.

Solución

- Cree un registro A para el dominio principal: debe crear un registro A en su configuración de DNS para el dominio principal de su dominio personalizado.
  - Ejemplo: si su dominio personalizado es `auth.xyz.yourdomain.com`, el dominio principal es `xyz.yourdomain.com`. Si quiere configurar `xyz.yourdomain.com` como un dominio personalizado, el dominio principal es `yourdomain.com`.
  - El dominio principal debe resolverse en una dirección IP válida. Si no apunta a una dirección IP real, puede usar una dirección IP ficticia, como `8.8.8.8`.
- Verifique la propagación del DNS (opcional pero recomendable): para asegurarse de que su proveedor de DNS haya propagado el cambio, puede ejecutar un comando `dig`.
  - Si usa `auth.xyz.yourdomain.com` como dominio personalizado: `dig A xyz.yourdomain.com +short`
  - Si usa `xyz.yourdomain.com` como dominio personalizado: `dig A yourdomain.com +short`
  - El comando debe devolver la dirección IP que configuró. Si no es así, espere a que el cambio se haya propagado por completo.
- Elimine el registro A del dominio principal: tras crear correctamente el dominio personalizado en Amazon Cognito, puede eliminar el registro A que creó para el dominio principal si era ficticio.

Para obtener más información, consulte Uso de un dominio propio con la IU alojada.

El dominio ya está asociado a otro grupo de usuarios

Los nombres de dominio personalizados deben ser únicos en todas Cuentas de AWS las regiones.

Problema

Los nombres de dominio personalizados deben ser únicos en todas Cuentas de AWS las regiones.

Solución

- Para usar el nombre de dominio para un nuevo grupo de usuarios, debe eliminar el dominio personalizado del grupo de usuarios al que está asociado actualmente.
- Espera después de la eliminación: el dominio personalizado tarda un tiempo en eliminarse por completo del primer grupo de usuarios. Es posible que se siga produciendo este error al crear un nuevo dominio personalizado con el mismo nombre inmediatamente después de eliminarlo. Espere unos minutos e inténtelo de nuevo.

Uno o más de los CNAME que has proporcionado ya están asociados a un recurso diferente

Al crear un dominio personalizado, Amazon Cognito crea una distribución de Amazon AWS CloudFront gestionada. Un nombre de dominio solo se puede utilizar con una CloudFront distribución de Amazon. Este error se produce si el nombre de dominio ya está en uso como dominio alternativo para otra CloudFront distribución de Amazon.

Problema

Al crear un dominio personalizado, Amazon Cognito crea una distribución de Amazon AWS CloudFront gestionada. Un nombre de dominio solo se puede utilizar con una CloudFront distribución de Amazon. Este error se produce si el nombre de dominio ya está en uso como dominio alternativo para otra CloudFront distribución de Amazon.

Solución

- Opción 1: use un nombre de dominio diferente para su dominio personalizado de Amazon Cognito.
- Opción 2: si utiliza el nombre de dominio de Amazon Cognito, no lo utilice con otra distribución de Amazon CloudFront .

El certificado SSL especificado no existe

Amazon Cognito utiliza Amazon CloudFront, que exige que el certificado AWS Gestor de certificados (ACM) esté en `us-east-1` (Virginia del Norte) Región de AWS, independientemente de la región del grupo de usuarios.

Problema

Amazon Cognito utiliza Amazon CloudFront, que exige que el certificado AWS Gestor de certificados (ACM) esté en `us-east-1` (Virginia del Norte) Región de AWS, independientemente de la región del grupo de usuarios.

Solución

- Compruebe la región del certificado: confirme que el certificado ACM se encuentra en la región `us-east-1`.
- Compruebe la validez del certificado: asegúrese de que el certificado seleccionado no esté caducado.
- Certificados importados. Si importó el certificado a ACM, compruebe lo siguiente:
  - Lo emitió una autoridad pública de certificación.
  - Incluye la cadena de certificados correcta.
- AWS KMS Comprobación de políticas: este error puede estar provocado por una `deny` declaración explícita en una política AWS Key Management Service (AWS KMS) del usuario o rol de IAM que crea el dominio. Sobre todo, compruebe si hay denegaciones explícitas en acciones `kms:DescribeKey`, `kms:CreateGrant` o `kms:*`.

Error de **`token de actualización no válido`**

Problema

Recibe un error `Invalid refresh token` al intentar utilizar un token de actualización para obtener nuevos tokens de acceso e ID de su grupo de usuarios de Amazon Cognito mediante la operación de API `AdminInitiateAuth` o `InitiateAuth`.

Solución

Implemente los siguientes pasos de resolución de problemas en función de la configuración de su grupo de usuarios y del uso de la API:

- Use el mismo ID de cliente de aplicación: al llamar a la API `AdminInitiateAuth` o `InitiateAuth` al actualizar el token, debe usar exactamente el mismo ID de cliente de aplicación que se utilizó durante la autenticación inicial que generó el token de actualización.
- Confirma el dispositivo: si tienes habilitado el seguimiento de dispositivos en tu grupo de usuarios, pero el dispositivo del usuario no se ha confirmado, primero debes llamar a la

ConfirmDeviceAPI. Cuando el usuario confirme su dispositivo, puede intercambiar el token de actualización.

- Incluya la clave del dispositivo en la solicitud de actualización: si el seguimiento del dispositivo está activado, incluya la clave única del dispositivo como `AuthParameter` cuando utilice el flujo `REFRESH_TOKEN_AUTH`:

```
{
  "AuthFlow": "REFRESH_TOKEN_AUTH",
  "AuthParameters": {
    "REFRESH_TOKEN": "example_refresh_token",
    "SECRET_HASH": "example_secret_hash", // Required if your app client uses a
 client secret
    "DEVICE_KEY": "example_device_key"
  }
}
```

- Use **`USER_SRP_AUTH`** para el seguimiento de dispositivos: si utiliza el seguimiento de dispositivos, el flujo de autenticación inicial debe ser `USER_SRP_AUTH`.

Para obtener más información, consulte Uso de dispositivos de usuario en el grupos de usuarios.

Errores de respuesta SAML no válidos en la federación

Los usuarios reciben varios errores `Invalid SAML response` similares al intentar federarse en Amazon Cognito mediante SAML 2.0. Estos errores pueden producirse debido a problemas de asignación de atributos, problemas con los certificados o discordancias de configuración.

`Atributos de usuario no válidos: atributo obligatorio`

Problema

A un usuario le falta un valor para un atributo que es obligatorio en su grupo de usuarios, o el IdP está intentando eliminar o actualizar un atributo inmutable.

Solución

- Compruebe los atributos obligatorios definidos en la configuración de su grupo de usuarios.
- Con las herramientas de captura de red de su navegador, recupere la respuesta SAML. Es posible que tenga que realizar una decodificación de URL y base64. Compruebe que el atributo esté en la aserción de SAML.

- Inicie sesión en su proveedor de identidad y revise los atributos que envía a Amazon Cognito. Compruebe que el IdP esté configurado para enviar el atributo necesario con el nombre correcto.
- Para los atributos inmutables, ejecute el siguiente AWS CLI comando para identificarlos:. `aws cognito-idp describe-user-pool --user-pool-id USER-POOL-ID --query 'UserPool.SchemaAttributes[?Mutable==`false`].Name'`
- En las asignaciones de atributos de SAML de su IdP, elimine todas las asignaciones que se dirijan a un atributo inmutable de Amazon Cognito. Si lo prefiere, actualice el atributo de destino a un atributo diferente y mutable.

Se recibió una respuesta SAML no válida: la firma de la respuesta SAML no es válida

Problema

Su IdP ha actualizado su certificado de firma de SAML, lo que provoca una discrepancia entre el certificado de la respuesta de SAML y el archivo de metadatos almacenado en Amazon Cognito.

Solución

1. Descargue el archivo de metadatos más reciente de su IdP.
2. En la consola de Amazon Cognito, vaya a los proveedores sociales y externos de su grupo de usuarios, edite su proveedor de SAML y sustituya el archivo de metadatos existente por el archivo recién descargado.

No se **encontró la restricción de audiencia** o **la aplicación con identificador**

Problema

Se ha configurado un ID de entidad incorrecto en su IdP o la aserción utiliza el nombre de recurso uniforme (URN) de otro grupo de usuarios.

Solución

1. Obtenga su ID de grupo de usuarios de Amazon Cognito en la sección Descripción general de la consola.

2. En la consola de administración de su IdP, actualice el ID de la entidad en la aplicación SAML de su grupo de usuarios. Configure el ID de la entidad para que coincida con el formato `urn:amazon:cognito:sp:`*`USER_POOL_ID`*. Sustituya `USER_POOL_ID` por el ID del grupo de usuarios del paso anterior.

Se ha encontrado un error en la página solicitada

Problema

La URL del Assertion Consumer Service (ACS) registrada con su IdP está mal configurada o el IdP no envía la respuesta SAML mediante la vinculación POST requerida.

Solución

- En la consola de administración de su IdP, actualice la aplicación con el formato de URL ACS correcto y compruebe que está usando la vinculación `HTTP POST`.
- Formato de dominio predeterminado: `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/saml2/idpresponse`
- Formato de dominio personalizado: `https://`*`auth.example.com`*`/saml2/idpresponse`

El RelayState del proveedor de identidad no es válido

Problema

Falta el parámetro `RelayState` o no es válido, o la URL del IdP y Amazon Cognito no coinciden.

Solución

- Para los flujos iniciados por el proveedor de servicios (SP-initiated): inicie siempre la autenticación en el punto final del grupo `/oauth2/authorize` de usuarios. La afirmación de SAML de que no devuelve `RelayState` parámetros de la visita inicial del usuario a Amazon Cognito no es una solicitud válida. SP-initiated
- Para los flujos iniciados por el proveedor de identidad (IdP-initiated): el IdP debe incluir el `RelayState` parámetro con la afirmación SAML en el `/saml2/idpresponse` punto final, utilizando el formato requerido:. `redirect_uri=REDIRECT_URI&state=STATE`

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

Los usuarios de inicio de sesión administrado no pueden seleccionar un factor MFA

Problema

Los usuarios no pueden elegir el método de MFA que prefieran al iniciar sesión mediante el inicio de sesión administrado o no se les solicita el método de MFA esperado.

Solución

Amazon Cognito sigue una lógica específica para determinar qué factores de MFA están disponibles para los usuarios en función de la configuración del grupo de usuarios, los atributos de los usuarios y la configuración de recuperación de la cuenta. Por ejemplo, los usuarios no pueden usar el MFA de correo electrónico si el correo electrónico está configurado como el método de recuperación de su cuenta principal.

Para anular la selección predeterminada de factores de MFA, puede implementar uno de estos procesos:

- Para aplicaciones públicas: úselo SetUserMFAPreferencecon un token de acceso válido para que los usuarios puedan establecer sus propias preferencias de MFA
- Para aplicaciones administradas por el administrador: utilícelas AdminSetUserMFAPreferencecon AWS credenciales para configurar las preferencias de MFA del usuario

Ambas operaciones le permiten habilitar o deshabilitar los métodos de SMS, correo electrónico y MFA TOTP para usuarios individuales, y configurar así un método como preferido.

Para obtener más información, consulte Adición de MFA a un grupo de usuarios..

Los usuarios del método sin contraseña y con clave de acceso no pueden usar la MFA

Problema

Los usuarios que inician sesión con métodos de autenticación con contraseña de un solo uso (OTP) no pueden agregar ni usar la autenticación multifactorial.

Solución

Los flujos de inicio de sesión de OTP no admiten MFA. Sin embargo, la autenticación con clave de paso con verificación de usuario puede cumplir con los requisitos de MFA.

`FactorConfiguration`Configúrelo `MULTI_FACTOR_WITH_USER_VERIFICATION` en su grupo de usuarios `WebAuthnConfiguration` para permitir que las claves de paso cuenten como autenticación multifactorial.

Para obtener más información, consulte Autenticación con grupos de usuarios.

No se puede recibir el código de restablecimiento de contraseña a través de email/SMS

Problema

Los usuarios no pueden recibir los códigos de verificación durante el proceso de olvido de la contraseña mediante su correo electrónico o SMS.

Solución

Hay varias razones por las que es posible que no se reciban los códigos de verificación. Siga esta lista de comprobación para solucionar el problema:

- Compruebe las carpetas de correo no deseado y spam del usuario.
- Confirme que el usuario existe en el grupo de usuarios.
- Compruebe que el estado del usuario no sea `FORCE_CHANGE_PASSWORD`. Si este es el caso, el usuario lo creó un administrador y Amazon Cognito le pedirá que establezca una contraseña cuando inicie sesión con su contraseña temporal.
- Compruebe que el usuario tenga un atributo de correo electrónico o número de teléfono verificados.
- Compruebe el límite de gasto de su cuenta para la mensajería SMS
- Consulte su cuota de envío de mensajes de Amazon Simple Email Service (Amazon SES).
- Compruebe si tanto SMS como Amazon SES (en el caso de que su grupo de usuarios esté configurado para enviar correo electrónico con Amazon SES) se han retirado del entorno de pruebas. Si no se han retirado del entorno de pruebas, las direcciones de correo electrónico o los números de teléfono que no hayan sido verificados por Amazon SES o AWS End User Messaging SMS que no puedan recibir códigos de restablecimiento de contraseñas.
- Si el usuario tiene un factor de MFA activo, compruebe que no está intentando generar un mensaje con el mismo factor. Por ejemplo, los usuarios con el MFA de correo electrónico activo no pueden enviar códigos de restablecimiento de contraseña por correo electrónico.

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.

El restablecimiento de la contraseña no se puede restablecer debido a los atributos de recuperación no verificados: **`no se ha podido restablecer la contraseña de la cuenta, póngase en contacto con el servicio de asistencia o`** inténtelo de nuevo

Problema

Los usuarios reciben este error cuando intentan restablecer su contraseña debido a métodos de recuperación no verificados o a conflictos de configuración de MFA. El error se produce cuando el atributo de correo electrónico o número de teléfono del usuario no está verificado o cuando su configuración de MFA impide utilizar el método de recuperación configurado.

Solución

Revisa la configuración de recuperación de cuentas de tu grupo de usuarios y el estado de verificación del usuario afectado:

- Compruebe la configuración de recuperación: en su grupo de usuarios, vaya a Iniciar sesión > Recuperación de cuentas de usuario. Asegúrese de que la recuperación automática de cuentas esté habilitada y revise la configuración del método de entrega de mensajes de recuperación.
- Compruebe la verificación de los atributos del usuario: compruebe que el usuario tiene una dirección de correo electrónico o un número de teléfono verificados que coincidan con la configuración del método de recuperación. En la consola, vaya al perfil del usuario, seleccione Atributos de usuario > Editar y marque el atributo correspondiente como verificado.
- Resolver conflictos de MFA: los usuarios cuyo método de MFA preferido sea el correo electrónico no pueden recibir códigos de restablecimiento de contraseñas por correo electrónico y los usuarios con MFA por SMS no pueden recibir códigos por SMS. Actualiza el método de entrega de los mensajes de recuperación para ofrecer opciones alternativas, como SMS si están disponibles, correo electrónico o correo electrónico si está disponible o SMS.
- Verificación administrativa: utilice la operación de la API AdminUpdateUserAttributespara verificar mediante programación los atributos del usuario cuando el acceso a la consola no esté disponible.

Para obtener más información, consulte Contraseñas, recuperación de contraseñas y políticas de contraseñas.

Errores de SECRET_HASH

Problema

Las solicitudes de la API de autenticación a los clientes de aplicación con secretos de cliente devuelven errores como `An error occurred (NotAuthorizedException) when calling the ForgotPassword operation: Client 1example23456789 is configured with secret but SECRET_HASH was not received.`

Solución

La aplicación debe calcular el `SECRET_HASH` para el usuario actual, el cliente de aplicación y el secreto del cliente. Este es el método de cálculo:

```
Base64 ( HMAC_SHA256 ( "client secret", "Username" + "Client Id" ) )
```

Para obtener el secreto del cliente, puede hacer lo siguiente:

1. Abra la consola de Amazon Cognito y diríjase a su cliente de aplicación desde el menú Clientes de aplicación. En la sección Información del cliente de aplicación, busque el Secreto de cliente. Seleccione Mostrar el secreto de cliente y aparecerá el secreto del cliente de la aplicación.
2. Genera una solicitud. DescribeUserPoolClient El secreto del cliente está incluido en la respuesta.

Para obtener más información, consulte Cálculo de los valores de hash secretos.

La consola de Amazon Cognito elige una configuración predeterminada para un nuevo grupo de usuarios

Problema

Al configurar un nuevo grupo de usuarios en la consola, Amazon Cognito selecciona varios ajustes predeterminados de forma automática. Algunos ajustes no se pueden cambiar una vez creado el grupo de usuarios. ¿Cómo puede tomar decisiones informadas y entender qué es lo que Amazon Cognito seleccionó automáticamente?

Solución

La nueva configuración del grupo de usuarios de la consola de Amazon Cognito está diseñada para realizar pruebas y crear prototipos rápidamente. La consola le presenta solo las opciones de configuración más importantes, aquellas que no se pueden cambiar tras la creación del grupo de usuarios. Todos los demás ajustes que Amazon Cognito configure automáticamente se pueden modificar más adelante.

Se recomienda el siguiente enfoque:

1. Utilice la consola para crear grupos de usuarios de prueba mientras perfecciona la implementación.
2. Una vez que haya determinado la configuración de producción, aplíquela a los grupos de usuarios de prueba.
3. Utilice DescribeUserPoollas operaciones de DescribeUserPoolClientAPI para generar plantillas JSON de la configuración probada.
4. Utilice estas plantillas con herramientas de implementación, como los AWS SDK, el CDK o la API REST, o CloudFormation para crear sus recursos de producción.

Para obtener más información, consulte Creación de una nueva aplicación en la consola de Amazon Cognito.

Recursos adicionales de solución de problemas

Para obtener orientación adicional sobre la resolución de problemas y las soluciones aportadas por la comunidad, también puede consultar los siguientes recursos externos:

- AWS Comunidad re:post de Amazon Cognito: explore las preguntas y soluciones de la comunidad
- AWS Centro de conocimiento: artículos de Amazon Cognito: artículos seleccionados sobre solución de problemas

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 de AWS SDK y ejemplos de código, consulte Uso de este servicio con un AWS SDK. 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 AWS SDKs
  - Ejemplos básicos de Amazon Cognito Identity mediante AWS SDKs
    - Acciones para el uso de Amazon Cognito Identity AWS SDKs
      - Úselo CreateIdentityPool con un AWS SDK o CLI
      - Úselo DeleteIdentityPool con un AWS SDK o CLI
      - Utilizar DescribeIdentityPool con una CLI
      - GetCredentialsForIdentityÚselo con un AWS SDK
      - Utilizar GetIdentityPoolRoles con una CLI
      - Úselo ListIdentityPools con un AWS SDK o CLI
      - Utilizar SetIdentityPoolRoles con una CLI
      - Utilizar UpdateIdentityPool con una CLI
  - Escenarios de uso de Amazon Cognito Identity AWS SDKs
    - Creación de una aplicación de exploración de Amazon Textract
- Ejemplos de código para Amazon Cognito Identity Provider mediante AWS SDK
  - Ejemplos básicos de Amazon Cognito Identity Provider mediante AWS SDK
    - Introducción a Amazon Cognito
    - Acciones para el proveedor de identidad de Amazon Cognito mediante AWS SDK
      - AdminCreateUserÚselo con un AWS SDK o CLI
      - AdminGetUserÚselo con un AWS SDK o CLI
      - AdminInitiateAuthÚselo con un AWS SDK o CLI
      - AdminRespondToAuthChallengeÚselo con un AWS SDK o CLI
      - AdminSetUserPasswordÚselo con un AWS SDK o CLI

- AssociateSoftwareTokenÚselo con un AWS SDK o CLI
- ConfirmDeviceÚselo con un AWS SDK o CLI
- ConfirmForgotPasswordÚselo con un AWS SDK o CLI
- ConfirmSignUpÚselo con un AWS SDK o CLI
- CreateUserPoolÚselo con un AWS SDK o CLI
- CreateUserPoolClientÚselo con un AWS SDK o CLI
- DeleteUserÚselo con un AWS SDK o CLI
- DescribeTermsByClientÚselo con un AWS SDK
- DescribeUserPoolClientÚselo con un AWS SDK o CLI
- ForgotPasswordÚselo con un AWS SDK o CLI
- GetClientTokenÚselo con un AWS SDK
- InitiateAuthÚselo con un AWS SDK o CLI
- ListUserPoolsÚselo con un AWS SDK o CLI
- ListUsersÚselo con un AWS SDK o CLI
- ResendConfirmationCodeÚselo con un AWS SDK o CLI
- RespondToAuthChallengeÚselo con un AWS SDK o CLI
- SignUpÚselo con un AWS SDK o CLI
- UpdateUserPoolÚselo con un AWS SDK o CLI
- VerifySoftwareTokenÚselo con un AWS SDK o CLI

- Escenarios en los que Amazon Cognito Identity Provider utiliza AWS SDK
  - Confirme automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un AWS SDK
  - Migre automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un AWS SDK
  - Obtenga tokens de cliente M2M y descubra las condiciones de Amazon Cognito mediante un AWS SDK
  - Introducción a los grupos de usuarios
  - Registrar un usuario con un grupo de usuarios de Amazon Cognito que requiera MFA mediante un AWS SDK
  - Uso de los grupos de identidades y los flujos de identidades de Amazon Cognito

        - Escriba datos de actividad personalizados con una función Lambda tras la autenticación de usuarios de Amazon Cognito mediante un AWS SDK
- Ejemplos de código para Amazon Cognito Sync mediante AWS SDKs
  - Ejemplos básicos de Amazon Cognito Sync mediante AWS SDKs
    - Acciones para Amazon Cognito Sync mediante AWS SDKs
      - ListIdentityPoolUsageÚselo con un AWS SDK

Ejemplos de código para Amazon Cognito Identity mediante AWS SDKs

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 distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros 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 AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Amazon Cognito Identity

- Ejemplos básicos de Amazon Cognito Identity mediante AWS SDKs
  - Acciones para el uso de Amazon Cognito Identity AWS SDKs
    - Úselo CreateIdentityPool con un AWS SDK o CLI
    - Úselo DeleteIdentityPool con un AWS SDK o CLI
    - Utilizar DescribeIdentityPool con una CLI
    - GetCredentialsForIdentityÚselo con un AWS SDK
    - Utilizar GetIdentityPoolRoles con una CLI
    - Úselo ListIdentityPools con un AWS SDK o CLI
    - Utilizar SetIdentityPoolRoles con una CLI

    - Utilizar UpdateIdentityPool con una CLI
- Escenarios de uso de Amazon Cognito Identity AWS SDKs
  - Creación de una aplicación de exploración de Amazon Textract

Ejemplos básicos de Amazon Cognito Identity mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de Amazon Cognito Identity con. AWS SDKs

Ejemplos

- Acciones para el uso de Amazon Cognito Identity AWS SDKs
  - Úselo CreateIdentityPool con un AWS SDK o CLI
  - Úselo DeleteIdentityPool con un AWS SDK o CLI
  - Utilizar DescribeIdentityPool con una CLI
  - GetCredentialsForIdentityÚselo con un AWS SDK
  - Utilizar GetIdentityPoolRoles con una CLI
  - Úselo ListIdentityPools con un AWS SDK o CLI
  - Utilizar SetIdentityPoolRoles con una CLI
  - Utilizar UpdateIdentityPool con una CLI

Acciones para el uso de Amazon Cognito Identity AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Cognito Identity con. AWS SDKs Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

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. Puede ver las acciones en contexto en Escenarios de uso de Amazon Cognito Identity AWS SDKs.

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
- Utilizar DescribeIdentityPool con una CLI
- GetCredentialsForIdentityÚselo con un AWS SDK
- Utilizar GetIdentityPoolRoles con una CLI
- Úselo ListIdentityPools con un AWS SDK o CLI
- Utilizar SetIdentityPoolRoles con una CLI
- Utilizar UpdateIdentityPool con una CLI

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

Los siguientes ejemplos de código muestran 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 la PowerShell versión 4

Ejemplo 1: Creación de 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 CreateIdentityPool Herramientas de AWS para PowerShellCmdlet Reference (V4).

Herramientas para la versión 5 PowerShell

Ejemplo 1: Creación de 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 CreateIdentityPoolla referencia de Herramientas de AWS para PowerShell cmdlets (V5).

Swift

SDK para Swift

> **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 AWSCognitoIdentity


    /// 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? {
        do {
            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
        } catch {
            print("ERROR: createIdentityPool:", dump(error))
            throw error
        }
    }
```

- 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 AWS SDK. 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

Los siguientes ejemplos de código muestran 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 la PowerShell versión 4

Ejemplo 1: Eliminación de un grupo de identidades específico.

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

- Para obtener más información sobre la API, consulte DeleteIdentityPool Herramientas de AWS para PowerShellCmdlet Reference (V4).

Herramientas para la versión 5 PowerShell

Ejemplo 1: Eliminación de un grupo de identidades específico.

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

- Para obtener más información sobre la API, consulte DeleteIdentityPoolla referencia de Herramientas de AWS para PowerShell cmdlets (V5).

Swift

SDK para Swift

> **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 AWSCognitoIdentity


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

            _ = try await cognitoIdentityClient.deleteIdentityPool(input: input)
        } catch {
            print("ERROR: deleteIdentityPool:", dump(error))
            throw error
        }
```

```
    }
```

- 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 AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Utilizar **`DescribeIdentityPool`** con una CLI

Los siguientes ejemplos de código muestran 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 la PowerShell versión 4

Ejemplo 1: Recuperación de información sobre un grupo de identidades específico mediante el 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 DescribeIdentityPool Herramientas de AWS para PowerShellCmdlet Reference (V4).

Herramientas para la versión 5 PowerShell

Ejemplo 1: Recuperación de información sobre un grupo de identidades específico mediante el 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 DescribeIdentityPoolla referencia de Herramientas de AWS para PowerShell cmdlets (V5).

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

`GetCredentialsForIdentity`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `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 AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Utilizar **`GetIdentityPoolRoles`** con una CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetIdentityPoolRoles`.

CLI

AWS CLI

Para obtener roles del grupo de identidades

En este ejemplo, se obtienen los roles de grupos 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 la PowerShell versión 4

Ejemplo 1: Obtención de información sobre los roles 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 GetIdentityPoolRoles Herramientas de AWS para PowerShellCmdlet Reference (V4).

Herramientas para la versión 5 PowerShell

Ejemplo 1: Obtención de información sobre los roles 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 GetIdentityPoolRolesla referencia de Herramientas de AWS para PowerShell cmdlets (V5).

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un AWS SDK 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

Los siguientes ejemplos de código muestran 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 la PowerShell versión 4

Ejemplo 1: Recuperación de 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 ListIdentityPools Herramientas de AWS para PowerShellCmdlet Reference (V4).

Herramientas para la versión 5 PowerShell

Ejemplo 1: Recuperación de 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 ListIdentityPoolsla referencia de Herramientas de AWS para PowerShell cmdlets (V5).

Swift

SDK para Swift

> **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 AWSCognitoIdentity


    /// 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? {
        let listPoolsInput = ListIdentityPoolsInput(maxResults: 25)
        // Use "Paginated" to get all the objects.
        // This lets the SDK handle the 'nextToken' field in
 "ListIdentityPoolsOutput".
        let pages = cognitoIdentityClient.listIdentityPoolsPaginated(input:
 listPoolsInput)

        do {
            for try await page in pages {
                guard let identityPools = page.identityPools else {
                    print("ERROR: listIdentityPoolsPaginated returned nil
 contents.")
                    continue
                }

                /// 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.

                for pool in identityPools {
                    if pool.identityPoolName == name {
                        return pool.identityPoolId!
                    }
                }
            }
        } catch {
            print("ERROR: getIdentityPoolID:", dump(error))
            throw error
        }

        return nil
```

```
    }
```

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

```
import AWSCognitoIdentity


    /// 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.

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

            return poolId
        } catch {
            print("ERROR: getOrCreateIdentityPoolID:", dump(error))
            throw error
        }
    }
```

- 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 AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Utilizar **`SetIdentityPoolRoles`** con una CLI

Los siguientes ejemplos de código muestran cómo utilizar `SetIdentityPoolRoles`.

CLI

AWS CLI

Para establecer roles del grupo de identidades

En el siguiente ejemplo de `set-identity-pool-roles`, se establecen roles para un 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 SetIdentityPoolRolesla Referencia de AWS CLI comandos.

PowerShell

Herramientas para la PowerShell versión 4

Ejemplo 1: Configuración del grupo de identidades específico para que tenga un rol de IAM no autenticado.

```
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 SetIdentityPoolRoles Herramientas de AWS para PowerShellCmdlet Reference (V4).

Herramientas para la versión 5 PowerShell

Ejemplo 1: Configuración del grupo de identidades específico para que tenga un rol de IAM no autenticado.

```
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 SetIdentityPoolRolesla referencia de Herramientas de AWS para PowerShell cmdlets (V5).

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

Utilizar **`UpdateIdentityPool`** con una CLI

Los siguientes ejemplos de código muestran 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 un proveedor de identidades. 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 UpdateIdentityPoolla Referencia de AWS CLI comandos.

PowerShell

Herramientas para la PowerShell versión 4

Ejemplo 1: Actualización de 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 UpdateIdentityPool Herramientas de AWS para PowerShellCmdlet Reference (V4).

Herramientas para la versión 5 PowerShell

Ejemplo 1: Actualización de 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 Herramientas de AWS para PowerShell cmdlets (V5).

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

Escenarios de uso de Amazon Cognito Identity AWS SDKs

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes en Amazon Cognito Identity con. AWS SDKs Estos escenarios muestran cómo realizar tareas específicas llamando a varias funciones dentro de la identidad de Amazon Cognito o en combinación con otros Servicios de AWS. En cada escenario se incluye un enlace al código fuente completo, con instrucciones de configuración y ejecución del código.

Los escenarios requieren un nivel intermedio de experiencia para entender las acciones de servicio en su contexto.

Ejemplos

- Creación de una aplicación de exploración de Amazon Textract

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 para 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

Python

SDK para Python (Boto3)

Muestra cómo utilizar Amazon Textract para detectar elementos de texto, formulario y tabla en una imagen de documento. AWS SDK para Python (Boto3) La imagen de entrada y la salida de Amazon Textract aparecen en una aplicación Tkinter que permite explorar los elementos detectados.

- Envía la imagen de un documento a Amazon Textract y explora el resultado de los elementos detectados.
- Envía imágenes directamente a Amazon Textract o mediante un bucket de Amazon Simple Storage Service (Amazon S3).
- Utilice la función asincrónica APIs para iniciar un trabajo que publique una notificación en un tema del Amazon Simple Notification Service (Amazon SNS) cuando se complete el trabajo.
- Consulta una cola de Amazon Simple Queue Service (Amazon SQS) en busca de un mensaje de finalización de trabajo y muestra los resultados.

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 AWS SDK. 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 AWS SDK

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 distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

Amazon Cognito Identity Provider

- Ejemplos básicos de Amazon Cognito Identity Provider mediante AWS SDK
  - Introducción a Amazon Cognito
  - Acciones para el proveedor de identidad de Amazon Cognito mediante AWS SDK
    - AdminCreateUserÚselo con un AWS SDK o CLI
    - AdminGetUserÚselo con un AWS SDK o CLI
    - AdminInitiateAuthÚselo con un AWS SDK o CLI
    - AdminRespondToAuthChallengeÚselo con un AWS SDK o CLI
    - AdminSetUserPasswordÚselo con un AWS SDK o CLI
    - AssociateSoftwareTokenÚselo con un AWS SDK o CLI
    - ConfirmDeviceÚselo con un AWS SDK o CLI
    - ConfirmForgotPasswordÚselo con un AWS SDK o CLI
    - ConfirmSignUpÚselo con un AWS SDK o CLI
    - CreateUserPoolÚselo con un AWS SDK o CLI
    - CreateUserPoolClientÚselo con un AWS SDK o CLI
    - DeleteUserÚselo con un AWS SDK o CLI
    - DescribeTermsByClientÚselo con un AWS SDK
    - DescribeUserPoolClientÚselo con un AWS SDK o CLI
    - ForgotPasswordÚselo con un AWS SDK o CLI
    - GetClientTokenÚselo con un AWS SDK
    - InitiateAuthÚselo con un AWS SDK o CLI
    - ListUserPoolsÚselo con un AWS SDK o CLI
    - ListUsersÚselo con un AWS SDK o CLI
    - ResendConfirmationCodeÚselo con un AWS SDK o CLI
    - RespondToAuthChallengeÚselo con un AWS SDK o CLI
    - SignUpÚselo con un AWS SDK o CLI
    - UpdateUserPoolÚselo con un AWS SDK o CLI

    - VerifySoftwareTokenÚselo con un AWS SDK o CLI
- Escenarios en los que Amazon Cognito Identity Provider utiliza AWS SDK
    - Confirme automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un AWS SDK
    - Migre automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un AWS SDK
    - Obtenga tokens de cliente M2M y descubra las condiciones de Amazon Cognito mediante un AWS SDK
    - Introducción a los grupos de usuarios
    - Registrar un usuario con un grupo de usuarios de Amazon Cognito que requiera MFA mediante un AWS SDK
    - Uso de los grupos de identidades y los flujos de identidades de Amazon Cognito
    - Escriba datos de actividad personalizados con una función Lambda tras la autenticación de usuarios de Amazon Cognito mediante un AWS SDK

Ejemplos básicos de Amazon Cognito Identity Provider mediante AWS SDK

En los siguientes ejemplos de código, se muestra cómo utilizar los fundamentos básicos del proveedor de identidades de Amazon Cognito con AWS SDK.

Ejemplos

- Introducción a Amazon Cognito
- Acciones para el proveedor de identidad de Amazon Cognito mediante AWS SDK
    - AdminCreateUserÚselo con un AWS SDK o CLI
    - AdminGetUserÚselo con un AWS SDK o CLI
    - AdminInitiateAuthÚselo con un AWS SDK o CLI
    - AdminRespondToAuthChallengeÚselo con un AWS SDK o CLI
    - AdminSetUserPasswordÚselo con un AWS SDK o CLI
    - AssociateSoftwareTokenÚselo con un AWS SDK o CLI
    - ConfirmDeviceÚselo con un AWS SDK o CLI
    - ConfirmForgotPasswordÚselo con un AWS SDK o CLI
    - ConfirmSignUpÚselo con un AWS SDK o CLI

- CreateUserPoolÚselo con un AWS SDK o CLI
- CreateUserPoolClientÚselo con un AWS SDK o CLI
- DeleteUserÚselo con un AWS SDK o CLI
- DescribeTermsByClientÚselo con un AWS SDK
- DescribeUserPoolClientÚselo con un AWS SDK o CLI
- ForgotPasswordÚselo con un AWS SDK o CLI
- GetClientTokenÚselo con un AWS SDK
- InitiateAuthÚselo con un AWS SDK o CLI
- ListUserPoolsÚselo con un AWS SDK o CLI
- ListUsersÚselo con un AWS SDK o CLI
- ResendConfirmationCodeÚselo con un AWS SDK o CLI
- RespondToAuthChallengeÚselo con un AWS SDK o CLI
- SignUpÚselo con un AWS SDK o CLI
- UpdateUserPoolÚselo con un AWS SDK o CLI
- VerifySoftwareTokenÚselo con un AWS SDK o CLI

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 archivo CMakeLists.txt CMake.

```
# 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, consulta ListUserPoolsla Referencia de AWS SDK para C++ 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.

```
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() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 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(ctx)
  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 para 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 para 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.

```
import boto3

# Create a Cognito Identity Provider client
cognitoidp = boto3.client("cognito-idp")

# Initialize a paginator for the list_user_pools operation
paginator = cognitoidp.get_paginator("list_user_pools")

# Create a PageIterator from the paginator
page_iterator = paginator.paginate(MaxResults=10)

# Initialize variables for pagination
user_pools = []

# Handle pagination
for page in page_iterator:
    user_pools.extend(page.get("UserPools", []))

# Print the list of user pools
print("User Pools for the account:")
if user_pools:
    for pool in user_pools:
        print(f"Name: {pool['Name']}, ID: {pool['Id']}")
else:
    print("No user pools found.")
```

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

Ruby

SDK para Ruby

> **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.

```
require 'aws-sdk-cognitoidentityprovider'
require 'logger'

# CognitoManager is a class responsible for managing AWS Cognito operations
# such as listing all user pools in the current AWS account.
class CognitoManager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all user pools associated with the AWS account.
  def list_user_pools
    paginator = @client.list_user_pools(max_results: 10)
    user_pools = []
    paginator.each_page do |page|
      user_pools.concat(page.user_pools)
    end

    if user_pools.empty?
      @logger.info('No Cognito user pools found.')
    else
      user_pools.each do |user_pool|
        @logger.info("User pool ID: #{user_pool.id}")
        @logger.info("User pool name: #{user_pool.name}")
        @logger.info("User pool status: #{user_pool.status}")
        @logger.info('---')
      end
    end
  end
end
```

```
if $PROGRAM_NAME == __FILE__
  cognito_client = Aws::CognitoIdentityProvider::Client.new
  manager = CognitoManager.new(cognito_client)
  manager.list_user_pools
end
```

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia AWS SDK para Ruby 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

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

Los siguientes ejemplos de código muestran cómo realizar acciones individuales del proveedor de identidad de Amazon Cognito con AWS los SDK. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

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. Puede ver las acciones en contexto en Escenarios en los que Amazon Cognito Identity Provider utiliza AWS SDK.

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

- AdminCreateUserÚselo con un AWS SDK o CLI
- AdminGetUserÚselo con un AWS SDK o CLI
- AdminInitiateAuthÚselo con un AWS SDK o CLI
- AdminRespondToAuthChallengeÚselo con un AWS SDK o CLI
- AdminSetUserPasswordÚselo con un AWS SDK o CLI
- AssociateSoftwareTokenÚselo con un AWS SDK o CLI
- ConfirmDeviceÚselo con un AWS SDK o CLI

- ConfirmForgotPasswordÚselo con un AWS SDK o CLI
- ConfirmSignUpÚselo con un AWS SDK o CLI
- CreateUserPoolÚselo con un AWS SDK o CLI
- CreateUserPoolClientÚselo con un AWS SDK o CLI
- DeleteUserÚselo con un AWS SDK o CLI
- DescribeTermsByClientÚselo con un AWS SDK
- DescribeUserPoolClientÚselo con un AWS SDK o CLI
- ForgotPasswordÚselo con un AWS SDK o CLI
- GetClientTokenÚselo con un AWS SDK
- InitiateAuthÚselo con un AWS SDK o CLI
- ListUserPoolsÚselo con un AWS SDK o CLI
- ListUsersÚselo con un AWS SDK o CLI
- ResendConfirmationCodeÚselo con un AWS SDK o CLI
- RespondToAuthChallengeÚselo con un AWS SDK o CLI
- SignUpÚselo con un AWS SDK o CLI
- UpdateUserPoolÚselo con un AWS SDK o CLI
- VerifySoftwareTokenÚselo con un AWS SDK o CLI

`AdminCreateUser`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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 los siguientes ejemplos de código:

- Introducción a los grupos de usuarios
- 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 ejemplo `admin-create-user`, 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

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(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &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 para 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`AdminGetUser`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .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 para 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 para 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.fromEnvironment { 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).

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Get information about a specific user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The Amazon Cognito Identity Provider client to use.
    ///   - userName: The user to retrieve information about.
    ///   - userPoolId: The user pool to search for the specified user.
    ///
    /// - Returns: `true` if the user's information was successfully
    ///   retrieved. Otherwise returns `false`.
    func adminGetUser(cipClient: CognitoIdentityProviderClient, userName: String,
                      userPoolId: String) async -> Bool {
        do {
            let output = try await cipClient.adminGetUser(
                input: AdminGetUserInput(
                    userPoolId: userPoolId,
                    username: userName
                )
            )

            guard let userStatus = output.userStatus else {
                print("*** Unable to get the user's status.")
                return false
            }

            print("User status: \(userStatus)")
            return true
        } catch {
            return false
        }
```

```
    }
```

- Para obtener más información sobre la API, consulta AdminGetUserla 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`AdminInitiateAuth`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .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 para C++ de la API.

CLI

AWS CLI

Inicio de sesión de un usuario como administrador

En el siguiente ejemplo de `admin-initiate-auth`, se inicia la sesión del usuario diego@example.com. Este ejemplo también incluye metadatos para la protección contra amenazas y ClientMetadata para los activadores Lambda. El usuario está configurado para MFA con TOTP y recibe el desafío de proporcionar un código desde la aplicación del autenticador antes de poder completar la autenticación.

```
aws cognito-idp admin-initiate-auth \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --auth-flow ADMIN_USER_PASSWORD_AUTH \
    --auth-parameters USERNAME=diego@example.com,PASSWORD="My@Example
$Password3!",SECRET_HASH=ExampleEncodedClientIdSecretAndUsername= \
```

```
    --context-data="{\"EncodedData\":\"abc123example\",\"HttpHeaders\":
[{\"headerName\":\"UserAgent\",\"headerValue\":\"Mozilla/5.0 (Windows NT
 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0\"}],\"IpAddress\":
\"192.0.2.1\",\"ServerName\":\"example.com\",\"ServerPath\":\"/login\"}" \
    --client-metadata="{\"MyExampleKey\": \"MyExampleValue\"}"
```

Salida:

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeExample...",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "MyAuthenticatorApp",
        "USER_ID_FOR_SRP": "diego@example.com"
    }
}
```

Para obtener más información, consulte Flujo de autenticación de administrador en la Guía para desarrolladores de Amazon Cognito.

- 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 para 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.fromEnvironment { 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).

SAP ABAP

SDK para SAP ABAP

> **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.

```
    TRY.
        " Set up authentication parameters
        DATA(lt_auth_params) = VALUE /aws1/
cl_cgpauthparamstype_w=>tt_authparameterstype(
          ( VALUE /aws1/cl_cgpauthparamstype_w=>ts_authparameterstype_maprow(
              key = 'USERNAME'
              value = NEW /aws1/cl_cgpauthparamstype_w( iv_user_name ) ) )
          ( VALUE /aws1/cl_cgpauthparamstype_w=>ts_authparameterstype_maprow(
              key = 'PASSWORD'
              value = NEW /aws1/cl_cgpauthparamstype_w( iv_password ) ) )
        ).
```

```
      " Add SECRET_HASH if provided
      IF iv_secret_hash IS NOT INITIAL.
        INSERT VALUE #(
          key = 'SECRET_HASH'
          value = NEW /aws1/cl_cgpauthparamstype_w( iv_secret_hash )
        ) INTO TABLE lt_auth_params.
      ENDIF.

      oo_result = lo_cgp->admininitiateauth(
        iv_userpoolid = iv_user_pool_id
        iv_clientid = iv_client_id
        iv_authflow = 'ADMIN_USER_PASSWORD_AUTH'
        it_authparameters = lt_auth_params
      ).

      DATA(lv_challenge) = oo_result->get_challengename( ).

      IF lv_challenge IS INITIAL.
        MESSAGE 'User successfully signed in.' TYPE 'I'.
      ELSE.
        MESSAGE |Authentication challenge required: { lv_challenge }.| TYPE
'I'.
      ENDIF.

    CATCH /aws1/cx_cgpusernotfoundex INTO DATA(lo_user_ex).
      MESSAGE |User { iv_user_name } not found.| TYPE 'E'.

    CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
      MESSAGE 'Not authorized. Check credentials.' TYPE 'E'.
  ENDTRY.
```

- Para obtener más información sobre la API, consulte AdminInitiateAuthla referencia sobre la API ABAP del AWS SDK para SAP.

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Begin an authentication session.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID to use.
    ///   - userName: The username to check.
    ///   - password: The user's password.
    ///   - userPoolId: The user pool to use.
    ///
    /// - Returns: The session token associated with this authentication
    ///   session.
    func initiateAuth(cipClient: CognitoIdentityProviderClient, clientId: String,
                      userName: String, password: String,
                      userPoolId: String) async -> String? {
        var authParams: [String: String] = [:]

        authParams["USERNAME"] = userName
        authParams["PASSWORD"] = password

        do {
            let output = try await cipClient.adminInitiateAuth(
                input: AdminInitiateAuthInput(
                    authFlow:
 CognitoIdentityProviderClientTypes.AuthFlowType.adminUserPasswordAuth,
                    authParameters: authParams,
                    clientId: clientId,
                    userPoolId: userPoolId
                )
            )
```

```
            guard let challengeName = output.challengeName else {
                print("*** Invalid response from the auth service.")
                return nil
            }

            print("=====> Response challenge is \(challengeName)")

            return output.session
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return nil
        } catch _ as UserNotConfirmedException {
            print("*** The user \(userName) has not been confirmed.")
            return nil
        } catch {
            print("*** An unexpected error occurred.")
            return nil
        }
    }
```

- Para obtener más información sobre la API, consulta AdminInitiateAuthla 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`AdminRespondToAuthChallenge`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .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 para C++ de la API.

CLI

AWS CLI

Respuesta a un desafío de autenticación

Hay muchas maneras de responder a los diferentes desafíos de autenticación, en función del flujo de autenticación, la configuración del grupo de usuarios y los ajustes de los usuarios. En el siguiente ejemplo de `admin-respond-to-auth-challenge`, se proporciona un código MFA con TOTP para diego@example.com y se completa el inicio de sesión. Este grupo de usuarios tiene activada la función de recordar dispositivos, por lo que el resultado de la autenticación también devuelve una nueva clave de dispositivo.

```
aws cognito-idp admin-respond-to-auth-challenge \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --challenge-name SOFTWARE_TOKEN_MFA \
```

```
    --challenge-
responses USERNAME=diego@example.com,SOFTWARE_TOKEN_MFA_CODE=000000 \
    --session AYABeExample...
```

Salida:

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-ExAmPlE1"
        }
    }
}
```

Para obtener más información, consulte Flujo de autenticación de administrador en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela 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.

```
    // 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 para 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.fromEnvironment { 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).

SAP ABAP

SDK para SAP ABAP

> **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.

```
  TRY.
      " Build challenge responses
```

```
        DATA(lt_challenge_responses) = VALUE /aws1/
cl_cgpchallengerspstyp00=>tt_challengeresponsestype(
          ( VALUE /aws1/cl_cgpchallengerspstyp00=>ts_challengerspstype_maprow(
              key = 'USERNAME'
              value = NEW /aws1/cl_cgpchallengerspstyp00( iv_user_name ) ) )
          ( VALUE /aws1/cl_cgpchallengerspstyp00=>ts_challengerspstype_maprow(
              key = 'SOFTWARE_TOKEN_MFA_CODE'
              value = NEW /aws1/cl_cgpchallengerspstyp00( iv_mfa_code ) ) )
        ).

        " Add SECRET_HASH if provided
        IF iv_secret_hash IS NOT INITIAL.
          INSERT VALUE #(
            key = 'SECRET_HASH'
            value = NEW /aws1/cl_cgpchallengerspstyp00( iv_secret_hash )
          ) INTO TABLE lt_challenge_responses.
        ENDIF.

        DATA(lo_result) = lo_cgp->adminrespondtoauthchallenge(
          iv_userpoolid = iv_user_pool_id
          iv_clientid = iv_client_id
          iv_challengename = 'SOFTWARE_TOKEN_MFA'
          it_challengeresponses = lt_challenge_responses
          iv_session = iv_session
        ).

        oo_auth_result = lo_result->get_authenticationresult( ).

        IF oo_auth_result IS BOUND.
          MESSAGE 'MFA challenge completed successfully.' TYPE 'I'.
        ELSE.
          " Another challenge might be required
          DATA(lv_next_challenge) = lo_result->get_challengename( ).
          MESSAGE |Additional challenge required: { lv_next_challenge }.| TYPE
 'I'.
        ENDIF.

      CATCH /aws1/cx_cgpcodemismatchex INTO DATA(lo_code_ex).
        MESSAGE 'Invalid MFA code provided.' TYPE 'E'.

      CATCH /aws1/cx_cgpexpiredcodeex INTO DATA(lo_expired_ex).
        MESSAGE 'MFA code has expired.' TYPE 'E'.

      CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
```

```
        MESSAGE 'Not authorized. Check MFA configuration.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte AdminRespondToAuthChallengela referencia sobre la API ABAP del AWS SDK para SAP.

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Respond to the authentication challenge received from Cognito after
    /// initiating an authentication session. This involves sending a current
    /// MFA code to the service.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - userName: The user's username.
    ///   - clientId: The app client ID.
    ///   - userPoolId: The user pool to sign into.
    ///   - mfaCode: The 6-digit MFA code currently displayed by the user's
    ///     authenticator.
    ///   - session: The authentication session to continue processing.
    func adminRespondToAuthChallenge(cipClient: CognitoIdentityProviderClient,
 userName: String,
                                     clientId: String, userPoolId: String,
 mfaCode: String,
                                     session: String) async {
        print("=====> SOFTWARE_TOKEN_MFA challenge is generated...")

        var challengeResponsesOb: [String: String] = [:]
        challengeResponsesOb["USERNAME"] = userName
```

```
        challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

        do {
            let output = try await cipClient.adminRespondToAuthChallenge(
                input: AdminRespondToAuthChallengeInput(
                    challengeName:
CognitoIdentityProviderClientTypes.ChallengeNameType.softwareTokenMfa,
                    challengeResponses: challengeResponsesOb,
                    clientId: clientId,
                    session: session,
                    userPoolId: userPoolId
                )
            )

            guard let authenticationResult = output.authenticationResult else {
                print("*** Unable to get authentication result.")
                return
            }

            print("=====> Authentication result (JWTs are redacted):")
            print(authenticationResult)
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user \(userName) has not been confirmed.")
            return
        } catch let error as NotAuthorizedException {
            print("*** Unauthorized access. Reason: \(error.properties.message ??
"<unknown>")")
        } catch {
            print("*** Error responding to the MFA challenge.")
            return
        }
    }
```

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`AdminSetUserPassword`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `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

CLI

AWS CLI

Configuración de una contraseña de usuario como administrador

En el siguiente ejemplo de `admin-set-user-password`, se establece de forma permanente la contraseña de diego@example.com.

```
aws cognito-idp admin-set-user-password \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com \
    --password MyExamplePassword1! \
    --permanent
```

Este comando no genera ninguna salida.

Para obtener más información, consulte Contraseñas, recuperación de contraseñas y políticas de contraseñas en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulte AdminSetUserPasswordla 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

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(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &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 para 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`AssociateSoftwareToken`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .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 para C++ de la API.

CLI

AWS CLI

Generación de una clave secreta para una aplicación de autenticación MFA

En el siguiente ejemplo de `associate-software-token`, se genera una clave privada TOTP para un usuario que ha iniciado sesión y ha recibido un token de acceso. La clave privada resultante se puede ingresar manualmente en una aplicación de autenticación o las aplicaciones pueden renderizarla como un código QR que el usuario puede escanear.

```
aws cognito-idp associate-software-token \
    --access-token eyJra456defEXAMPLE
```

Salida:

```
{
    "SecretCode": "QWERTYUIOP123456EXAMPLE"
}
```

Para obtener más información, consulte MFA con token de software TOTP en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla 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 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 para 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.fromEnvironment { 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).

SAP ABAP

SDK para SAP ABAP

> **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.

```
TRY.
    DATA(lo_result) = lo_cgp->associatesoftwaretoken(
      iv_session = iv_session
    ).

    ov_secret_code = lo_result->get_secretcode( ).

    MESSAGE 'MFA secret code generated successfully.' TYPE 'I'.

  CATCH /aws1/cx_cgpresourcenotfoundex INTO DATA(lo_ex).
    MESSAGE 'Session not found or expired.' TYPE 'E'.

  CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
    MESSAGE 'Not authorized to associate software token.' TYPE 'E'.
ENDTRY.
```

- Para obtener más información sobre la API, consulte AssociateSoftwareTokenla referencia sobre la API ABAP del AWS SDK para SAP.

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Request and display an MFA secret token that the user should enter
    /// into their authenticator to set it up for the user account.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - authSession: The authentication session to request an MFA secret
    ///     for.
    ///
    /// - Returns: A string containing the MFA secret token that should be
    ///   entered into the authenticator software.
    func getSecretForAppMFA(cipClient: CognitoIdentityProviderClient,
 authSession: String?) async -> String? {
        do {
            let output = try await cipClient.associateSoftwareToken(
                input: AssociateSoftwareTokenInput(
                    session: authSession
                )
            )

            guard let secretCode = output.secretCode else {
                print("*** Unable to get the secret code")
                return nil
            }

            print("=====> Enter this token into Google Authenticator:
 \(secretCode)")
            return output.session
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
```

```
            return nil
        } catch {
            print("*** An unexpected error occurred getting the secret for the
app's MFA.")
            return nil
        }
    }
```

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ConfirmDevice`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .NET de la API.

CLI

AWS CLI

Confirmación del dispositivo de un usuario

En el siguiente ejemplo de `confirm-device`, se agrega un nuevo dispositivo recordado para el usuario actual.

```
aws cognito-idp confirm-device \
   --access-token eyJra456defEXAMPLE \
   --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
   --device-secret-verifier-
config PasswordVerifier=TXlWZXJpZmllclN0cmluZw,Salt=TXlTUlBTYWx0
```

Salida:

```
{
```

```
    "UserConfirmationNecessary": false
}
```

Para obtener más información, consulte Trabajo con dispositivos de usuario en el grupo de usuarios en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulta ConfirmDevicela 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 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 para 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ConfirmForgotPassword`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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 del 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &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 para 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ConfirmSignUp`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .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 para 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 para 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.fromEnvironment { 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).

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Submit a confirmation code for the specified user. This is the code as
    /// entered by the user after they've received it by email or text
    /// message.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID the user is signing up for.
    ///   - userName: The username of the user whose code is being sent.
    ///   - code: The user's confirmation code.
    ///
    /// - Returns: `true` if the code was successfully confirmed; otherwise
 `false`.
    func confirmSignUp(cipClient: CognitoIdentityProviderClient, clientId:
 String,
                       userName: String, code: String) async -> Bool {
        do {
            _ = try await cipClient.confirmSignUp(
                input: ConfirmSignUpInput(
                    clientId: clientId,
                    confirmationCode: code,
                    username: userName
                )
            )

            print("=====> \(userName) has been confirmed.")
            return true
        } catch {
            print("=====> \(userName)'s code was entered incorrectly.")
            return false
```

```
        }
    }
```

- Para obtener más información sobre la API, consulta ConfirmSignUpla 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateUserPool`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateUserPool`.

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:

- Introducción a los grupos de usuarios

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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`CreateUserPoolClient`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateUserPoolClient`.

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:

- Introducción a los grupos de usuarios

CLI

AWS CLI

Para crear un cliente de un grupo de usuarios

En el siguiente `create-user-pool-client` ejemplo, se crea un nuevo cliente de grupo de usuarios con un secreto de cliente, atributos de lectura y escritura explícitos, se inicia sesión con el nombre de usuario, la contraseña y los flujos de SRP, se inicia sesión con tres, se accede a un subconjunto de ámbitos de OAuth IdPs, se analiza y se amplía la validez de la sesión de autenticación. PinPoint

```
aws cognito-idp create-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-name MyTestClient \
```

```
    --generate-secret \
    --refresh-token-validity 10 \
    --access-token-validity 60 \
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days
 \
    --read-attributes email phone_number email_verified phone_number_verified \
    --write-attributes email phone_number \
    --explicit-auth-
flows ALLOW_USER_PASSWORD_AUTH ALLOW_USER_SRP_AUTH ALLOW_REFRESH_TOKEN_AUTH \
    --supported-identity-providers Google Facebook MyOIDC \
    --callback-urls https://www.amazon.com https://example.com http://
localhost:8001 myapp://example \
    --allowed-o-auth-flows code implicit \
    --allowed-o-auth-scopes openid profile aws.cognito.signin.user.admin solar-
system-data/asteroids.add \
    --allowed-o-auth-flows-user-pool-client \
    --analytics-configuration ApplicationArn=arn:aws:mobiletargeting:us-
west-2:767671399759:apps/thisisanexamplepinpointapplicationid,UserDataShared=TRUE
 \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
    --enable-propagate-additional-user-context-data \
    --auth-session-validity 4
```

Salida:

```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "MyTestClient",
        "ClientId": "123abc456defEXAMPLE",
        "ClientSecret": "this1234is5678my91011example1213client1415secret",
        "LastModifiedDate": 1726788459.464,
        "CreationDate": 1726788459.464,
        "RefreshTokenValidity": 10,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
```

```
        "ReadAttributes": [
            "email_verified",
            "phone_number_verified",
            "phone_number",
            "email"
        ],
        "WriteAttributes": [
            "phone_number",
            "email"
        ],
        "ExplicitAuthFlows": [
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
        ],
        "SupportedIdentityProviders": [
            "Google",
            "MyOIDC",
            "Facebook"
        ],
        "CallbackURLs": [
            "https://example.com",
            "https://www.amazon.com",
            "myapp://example",
            "http://localhost:8001"
        ],
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile",
            "solar-system-data/asteroids.add"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "AnalyticsConfiguration": {
            "ApplicationArn": "arn:aws:mobiletargeting:us-
west-2:123456789012:apps/thisisanexamplepinpointapplicationid",
            "RoleArn": "arn:aws:iam::123456789012:role/aws-service-role/cognito-
idp.amazonaws.com/AWSServiceRoleForAmazonCognitoIdp",
            "UserDataShared": true
        },
```

```
        "PreventUserExistenceErrors": "ENABLED",
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": true,
        "AuthSessionValidity": 4
    }
}
```

Para obtener más información, consulte la Application-specific configuración con clientes de aplicaciones en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulte CreateUserPoolClientla 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.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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DeleteUser`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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 para 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

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

```
// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &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 para 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.

```
/**
 * Delete the signed-in user. Useful for allowing a user to delete their
 * own profile.
 * @param {{ region: string, accessToken: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").DeleteUserCommandOutput | null, unknown]>}
 */
export const deleteUser = async ({ region, accessToken }) => {
  try {
    const client = new CognitoIdentityProviderClient({ region });
    const response = await client.send(
```

```
      new DeleteUserCommand({ AccessToken: accessToken }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

- Para obtener más información sobre la API, consulta DeleteUserla Referencia AWS SDK para JavaScript 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DescribeTermsByClient`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeTermsByClient`.

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:

- Descubrimiento de términos y fichas de cliente

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.

```
    def describe_terms_by_client(
        self, user_pool_id: str, client_id: str, terms_name: str
    ) -> dict[str, Any]:
        """
        Looks up a Terms document associated with an app client by client ID,
```

```
        user pool ID, and terms name.

        :param user_pool_id: The ID of the user pool that contains the Terms
documents.
        :param client_id: The ID of the app client associated with the Terms.
        :param terms_name: The name of the terms document ('terms-of-use' or
                           'privacy-policy').
        :return: A dictionary with the Terms document details.
        """
        try:
            response = self.cognito_idp_client.describe_terms_by_client(
                UserPoolId=user_pool_id,
                ClientId=client_id,
                TermsName=terms_name,
            )
            terms = response.get("Terms", dict())
            logger.info(
                "Described terms '%s' for client %s in user pool %s.",
                terms_name,
                client_id,
                user_pool_id,
            )
            return terms
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error(
                    "No terms documents found for client %s, user pool %s, "
                    "and terms name '%s'.",
                    client_id,
                    user_pool_id,
                    terms_name,
                )
            raise
```

- Para obtener más información sobre la API, consulta DescribeTermsByClientla 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DescribeUserPoolClient`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeUserPoolClient`.

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:

- Descubrimiento de términos y fichas de cliente
- Introducción a los grupos de usuarios

CLI

AWS CLI

Descripción de un cliente del grupo de usuarios

En este ejemplo, se describe a un cliente del grupo de usuarios.

Comando:

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --
client-id 38fjsnc484p94kpqsnet7mpld0
```

Salida:

```
{
  "UserPoolClient": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientName": "MyApp",
      "ClientId": "38fjsnc484p94kpqsnet7mpld0",
      "ClientSecret": "CLIENT_SECRET",
      "LastModifiedDate": 1548108676.163,
      "CreationDate": 1548108676.163,
      "RefreshTokenValidity": 30,
      "ReadAttributes": [
          "address",
          "birthdate",
          "custom:CustomAttr1",
          "custom:CustomAttr2",
          "email",
          "email_verified",
          "family_name",
```

```
            "gender",
            "given_name",
            "locale",
            "middle_name",
            "name",
            "nickname",
            "phone_number",
            "phone_number_verified",
            "picture",
            "preferred_username",
            "profile",
            "updated_at",
            "website",
            "zoneinfo"
        ],
        "WriteAttributes": [
            "address",
            "birthdate",
            "custom:CustomAttr1",
            "custom:CustomAttr2",
            "email",
            "family_name",
            "gender",
            "given_name",
            "locale",
            "middle_name",
            "name",
            "nickname",
            "phone_number",
            "picture",
            "preferred_username",
            "profile",
            "updated_at",
            "website",
            "zoneinfo"
        ],
        "ExplicitAuthFlows": [
            "ADMIN_NO_SRP_AUTH",
            "USER_PASSWORD_AUTH"
        ],
        "AllowedOAuthFlowsUserPoolClient": false
    }
}
```

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

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.

```
    def describe_user_pool_client(
        self, user_pool_id: str, client_id: str
    ) -> dict[str, Any]:
        """
        Describes an app client in a user pool, including the client secret.

        :param user_pool_id: The ID of the user pool that contains the app
client.
        :param client_id: The ID of the app client to describe.
        :return: A dictionary with the app client configuration.
        """
        try:
            response = self.cognito_idp_client.describe_user_pool_client(
                UserPoolId=user_pool_id, ClientId=client_id
            )
            client_info = response.get("UserPoolClient", dict())
            logger.info(
                "Described app client %s in user pool %s.",
                client_id,
                user_pool_id,
            )
            return client_info
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error(
                    "The user pool %s or client %s does not exist. "
                    "Verify the stack deployed successfully and the outputs are
correct.",
```

```
                    user_pool_id,
                    client_id,
                )
            raise
```

- Para obtener más información sobre la API, consulta DescribeUserPoolClientla 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ForgotPassword`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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 obligar a cambiar de contraseña

En el siguiente ejemplo de `forgot-password`, se envía un mensaje a jane@example.com para cambiar la 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

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(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &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 para 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`GetClientToken`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `GetClientToken`.

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:

- Descubrimiento de términos y fichas de cliente

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.

```
    def get_client_token(
        self,
```

```
        client_id: str,
        client_secret: str,
        scopes: Optional[list[str]] = None,
    ) -> dict[str, Any]:
        """
        Obtains an M2M access token for an app client using its client ID
        and secret, without end-user sign-in.

        :param client_id: The ID of the app client.
        :param client_secret: The client secret for the app client.
        :param scopes: Optional list of scopes in the format
                       'resource-server-identifier/scope-name'. If omitted,
                       all scopes configured for the app client are authorized.
        :return: A dictionary containing the ClientAuthenticationResult with
                 AccessToken, ExpiresIn, and TokenType.
        """
        try:
            params = dict()
            params["ClientId"] = client_id
            params["Secret"] = client_secret
            if scopes is not None:
                params["Scopes"] = scopes

            response = self.cognito_idp_client.get_client_token(**params)
            auth_result = response.get("ClientAuthenticationResult", dict())
            logger.info(
                "Obtained M2M access token for client %s. Token type: %s, expires
in: %s seconds.",
                client_id,
                auth_result.get("TokenType", "Unknown"),
                auth_result.get("ExpiresIn", 0),
            )
            return auth_result
        except ClientError as err:
            if err.response["Error"]["Code"] == "NotAuthorizedException":
                logger.error(
                    "Not authorized for client %s. Verify that the client ID and
"
                    "secret are correct, and that the ALLOW_CLIENT_TOKEN_AUTH
flow "
                    "is enabled for the app client.",
                    client_id,
                )
            raise
```

- Para obtener más información sobre la API, consulta GetClientTokenla 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`InitiateAuth`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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
- Introducción a los grupos de usuarios
- 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

SDK para .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 para .NET de la API.

CLI

AWS CLI

Inicio de sesión de un usuario

En el siguiente ejemplo de `initiate-auth`, se inicia la sesión de un usuario con el flujo básico de nombre de usuario y contraseña, sin desafíos adicionales.

```
aws cognito-idp initiate-auth \
    --auth-flow USER_PASSWORD_AUTH \
```

```
    --client-id 1example23456789 \
    --analytics-metadata AnalyticsEndpointId=d70b2ba36a8c4dc5a04a0451aEXAMPLE \
    --auth-parameters USERNAME=testuser,PASSWORD=[Password] --user-context-
data EncodedData=mycontextdata --client-metadata MyTestKey=MyTestValue
```

Salida:

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

Para obtener más información, consulte Autenticación en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulta InitiateAuthla 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.

```
import (
 "context"
```

```
 "errors"
 "log"

 "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"
)

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(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &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 para 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 para 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ListUserPools`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListUserPools`.

.NET

SDK para .NET (v4)

> **Note**
>
> Hay más información 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 para .NET de la API.

CLI

AWS CLI

Para mostrar los grupos de usuarios

En el siguiente `list-user-pools` ejemplo, se enumeran 3 de los grupos de usuarios disponibles en la AWS cuenta de las credenciales de CLI actuales.

```
aws cognito-idp list-user-pools \
    --max-results 3
```

Salida:

```
{
    "NextToken": "[Pagination token]",
    "UserPools": [
        {
            "CreationDate": 1681502497.741,
            "Id": "us-west-2_EXAMPLE1",
            "LambdaConfig": {
                "CustomMessage": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreSignUp": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGeneration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGenerationConfig": {
                    "LambdaArn": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                    "LambdaVersion": "V1_0"
                }
            },
            "LastModifiedDate": 1681502497.741,
            "Name": "user pool 1"
        },
        {
            "CreationDate": 1686064178.717,
            "Id": "us-west-2_EXAMPLE2",
            "LambdaConfig": {
            },
            "LastModifiedDate": 1686064178.873,
            "Name": "user pool 2"
```

```
        },
        {
            "CreationDate": 1627681712.237,
            "Id": "us-west-2_EXAMPLE3",
            "LambdaConfig": {
                "UserMigration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction"
            },
            "LastModifiedDate": 1678486942.479,
            "Name": "user pool 3"
        }
    ]
}
```

Para obtener más información, consulte Grupos de usuarios de Amazon Cognito en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulte 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() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 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(ctx)
  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 para 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ListUsers`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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 los siguientes ejemplos de código:

- Introducción a los grupos de usuarios
- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

SDK para .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 para .NET de la API.

CLI

AWS CLI

Ejemplo 1: muestra de usuarios con un filtro del servidor

En el siguiente ejemplo de `list-users`, se muestran 3 usuarios del grupo de usuarios solicitado cuyas direcciones de correo electrónico comienzan por `testuser`.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --filter email^=\"testuser\" \
    --max-items 3
```

Salida:

```
{
    "PaginationToken": "efgh5678EXAMPLE",
    "Users": [
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "eaad0219-2117-439f-8d46-4db20e59268f"
                },
                {
                    "Name": "email",
                    "Value": "testuser@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1682955829.578,
            "UserLastModifiedDate": 1689030181.63,
            "UserStatus": "CONFIRMED",
            "Username": "testuser"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "3b994cfd-0b07-4581-be46-3c82f9a70c90"
                },
                {
```

```
                    "Name": "email",
                    "Value": "testuser2@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427979.201,
            "UserLastModifiedDate": 1684427979.201,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser2"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "5929e0d1-4c34-42d1-9b79-a5ecacfe66f7"
                },
                {
                    "Name": "email",
                    "Value": "testuser3@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427823.641,
            "UserLastModifiedDate": 1684427823.641,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser3@example.com"
        }
    ]
}
```

Para obtener más información, consulte Administración y búsqueda de usuarios en la Guía para desarrolladores de Amazon Cognito.

Ejemplo 2: muestra de usuarios con un filtro del cliente

En el siguiente ejemplo de `list-users`, se muestran los atributos de tres usuarios que tienen un atributo, en este caso su dirección de correo electrónico, que contiene el dominio de correo electrónico "@example.com". Si otros atributos contuvieran esta cadena, también se mostrarían. El segundo usuario no tiene atributos que coincidan con la consulta y se excluye del resultado mostrado, pero no de la respuesta del servidor.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
```

```
    --max-items 3
    --query Users\[\*\].Attributes\[\?Value\.contains\(\@\,\'@example.com\'\)\]
```

Salida:

```
[
    [
        {
            "Name": "email",
            "Value": "admin@example.com"
        }
    ],
    [],
    [
        {
            "Name": "email",
            "Value": "operator@example.com"
        }
    ]
]
```

Para obtener más información, consulte Administración y búsqueda de usuarios en la Guía para desarrolladores de Amazon Cognito.

- 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 para 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.fromEnvironment { 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 ListUsersla AWS Referencia de API de SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **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.

```
    TRY.
        DATA(lo_result) = lo_cgp->listusers(
          iv_userpoolid = iv_user_pool_id
        ).

        ot_users = lo_result->get_users( ).

        MESSAGE |Found { lines( ot_users ) } users in the pool.| TYPE 'I'.

      CATCH /aws1/cx_cgpresourcenotfoundex INTO DATA(lo_ex).
        MESSAGE |User pool { iv_user_pool_id } not found.| TYPE 'E'.

      CATCH /aws1/cx_cgpnotauthorizedex INTO DATA(lo_auth_ex).
        MESSAGE 'Not authorized to list users.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte ListUsersla referencia sobre la API ABAP del AWS SDK para SAP.

Swift

SDK para Swift

> **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.

```
        do {
            let output = try await cognitoClient.listUsers(
                input: ListUsersInput(
                    userPoolId: poolId
                )
            )

            guard let users = output.users else {
                print("No users found.")
                return
            }
```

```
            print("\(users.count) user(s) found.")
            for user in users {
                print("  \(user.username ?? "<unknown>")")
            }
        } catch _ as NotAuthorizedException {
            print("*** Please authenticate with AWS before using this command.")
            return
        } catch _ as ResourceNotFoundException {
            print("*** The specified User Pool was not found.")
            return
        } catch {
            print("*** An unexpected type of error occurred.")
            return
        }
```

- Para obtener más información sobre la API, consulta ListUsersla 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`ResendConfirmationCode`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .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 para 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 para 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.fromEnvironment { 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).

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Requests a new confirmation code be sent to the given user's contact
    /// method.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The application client ID.
    ///   - userName: The user to resend a code for.
    ///
    /// - Returns: `true` if a new code was sent successfully, otherwise
    ///   `false`.
    func resendConfirmationCode(cipClient: CognitoIdentityProviderClient,
 clientId: String,
```

```
                                userName: String) async -> Bool {
        do {
            let output = try await cipClient.resendConfirmationCode(
                input: ResendConfirmationCodeInput(
                    clientId: clientId,
                    username: userName
                )
            )

            guard let deliveryMedium = output.codeDeliveryDetails?.deliveryMedium
else {
                print("*** Unable to get the delivery method for the resent
code.")
                return false
            }

            print("=====> A new code has been sent by \(deliveryMedium)")
            return true
        } catch {
            print("*** Unable to resend the confirmation code to user
\(userName).")
            return false
        }
    }
```

- Para obtener más información sobre la API, consulta ResendConfirmationCodela 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`RespondToAuthChallenge`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

Ejemplo 1: respuesta a un desafío NEW_PASSWORD_REQUIRED

En el siguiente ejemplo de `respond-to-auth-challenge`, se responde a un desafío NEW_PASSWORD_REQUIRED devuelto por initiate-auth. Establece una contraseña para el usuario `jane@example.com`.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --challenge-name NEW_PASSWORD_REQUIRED \
    --challenge-responses USERNAME=jane@example.com,NEW_PASSWORD=[Password] \
    --session AYABeEv5HklEXAMPLE
```

Salida:

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "ACCESS_TOKEN",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "REFRESH_TOKEN",
        "IdToken": "ID_TOKEN",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-wt2ha1Zd"
        }
    }
}
```

Para obtener más información, consulte Autenticación en la Guía para desarrolladores de Amazon Cognito.

Ejemplo 2: respuesta a un desafío SELECT_MFA_TYPE

En el siguiente ejemplo de `respond-to-auth-challenge`, se elige TOTP MFA como opción de MFA para el usuario actual. Se solicitó al usuario que seleccionara un tipo de MFA y, a continuación, se le pedirá que ingrese su código de MFA.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789
    --session AYABeEv5HklEXAMPLE
    --challenge-name SELECT_MFA_TYPE
    --challenge-responses USERNAME=testuser,ANSWER=SOFTWARE_TOKEN_MFA
```

Salida:

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeEv5HklEXAMPLE",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "transparent"
    }
}
```

Para obtener más información, consulte Agregación de MFA en la Guía para desarrolladores de Amazon Cognito.

Ejemplo 3: respuesta a un desafío SOFTWARE_TOKEN_MFA

En el siguiente ejemplo de `respond-to-auth-challenge`, se proporciona un código MFA con TOTP y se completa el inicio de sesión.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --session AYABeEv5HklEXAMPLE \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=testuser,SOFTWARE_TOKEN_MFA_CODE=123456
```

Salida:

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
```

```
        }
    }
}
```

Para obtener más información, consulte Agregación de MFA en la Guía para desarrolladores de Amazon Cognito.

- 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 para 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`SignUp`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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
- Introducción a los grupos de usuarios
- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

SDK para .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 para .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 para 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &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 para 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 para 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.fromEnvironment { 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).

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Create a new user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The ID of the app client to create a user for.
    ///   - userName: The username for the new user.
    ///   - password: The new user's password.
    ///   - email: The new user's email address.
    ///
    /// - Returns: `true` if successful; otherwise `false`.
    func signUp(cipClient: CognitoIdentityProviderClient, clientId: String,
 userName: String, password: String, email: String) async -> Bool {
        let emailAttr = CognitoIdentityProviderClientTypes.AttributeType(
            name: "email",
            value: email
        )

        let userAttrsList = [emailAttr]

        do {
            _ = try await cipClient.signUp(
                input: SignUpInput(
                    clientId: clientId,
                    password: password,
                    userAttributes: userAttrsList,
```

```
                    username: userName
                )

            )

            print("=====> User \(userName) signed up.")
        } catch _ as AWSCognitoIdentityProvider.UsernameExistsException {
            print("*** The username \(userName) already exists. Please use a
different one.")
            return false
        } catch let error as AWSCognitoIdentityProvider.InvalidPasswordException
{
            print("*** Error: The specified password is invalid. Reason:
\(error.properties.message ?? "<none available>").")
            return false
        } catch _ as AWSCognitoIdentityProvider.ResourceNotFoundException {
            print("*** Error: The specified client ID (\(clientId)) doesn't
exist.")
            return false
        } catch {
            print("*** Unexpected error: \(error)")
            return false
        }

        return true
    }
```

- Para obtener más información sobre la API, consulta SignUpla 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`UpdateUserPool`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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 el siguiente ejemplo de `update-user-pool`, se modifica un grupo de usuarios con un ejemplo de sintaxis para cada una de las opciones de configuración disponibles. Para actualizar un grupo de usuarios, debe especificar todas las opciones configuradas previamente o estas se restablecerán a su valor predeterminado.

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_EXAMPLE \
    --policies PasswordPolicy=
\{MinimumLength=6,RequireUppercase=true,RequireLowercase=true,RequireNumbers=true,Require
 \
    --deletion-protection ACTIVE \
    --lambda-config PreSignUp="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-presignup-
function",PreTokenGeneration="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-pretoken-function" \
    --auto-verified-attributes "phone_number" "email" \
    --verification-message-template \{\"SmsMessage\":\""Your code is
 {####}"\",\"EmailMessage\":\""Your code is {####}"\",\"EmailSubject\":\""Your
 verification code"\",\"EmailMessageByLink\":\""Click {##here##} to verify
 your email address."\",\"EmailSubjectByLink\":\""Your verification link"\",
\"DefaultEmailOption\":\"CONFIRM_WITH_LINK\"\} \
    --sms-authentication-message "Your code is {####}" \
    --user-attribute-update-settings
 AttributesRequireVerificationBeforeUpdate="email","phone_number" \
    --mfa-configuration "OPTIONAL" \
    --device-
configuration ChallengeRequiredOnNewDevice=true,DeviceOnlyRememberedOnUserPrompt=true
 \
    --email-configuration SourceArn="arn:aws:ses:us-
west-2:123456789012:identity/admin@example.com",ReplyToEmailAddress="amdin
```

```
+noreply@example.com",EmailSendingAccount=DEVELOPER,From="admin@amazon.com",Configuration
configuration-set" \
    --sms-configuration SnsCallerArn="arn:aws:iam::123456789012:role/service-
role/SNS-SMS-Role",ExternalId="12345",SnsRegion="us-west-2" \
    --admin-create-user-config
 AllowAdminCreateUserOnly=false,InviteMessageTemplate=\{SMSMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailSubject=\""Welcome to
 MyMobileGame"\"\} \
    --user-pool-tags "Function"="MyMobileGame","Developers"="Berlin" \
    --admin-create-user-config
 AllowAdminCreateUserOnly=false,InviteMessageTemplate=\{SMSMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailMessage=\""Welcome
 {username}. Your confirmation code is {####}"\",EmailSubject=\""Welcome to
 MyMobileGame"\"\} \
    --user-pool-add-ons AdvancedSecurityMode="AUDIT" \
    --account-recovery-setting RecoveryMechanisms=
\[\{Priority=1,Name="verified_email"\},
\{Priority=2,Name="verified_phone_number"\}\]
```

Este comando no genera ninguna salida.

Para obtener más información, consulte Updating user pool configuration en la Guía para desarrolladores de Amazon Cognito.

- 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.

```
import (
 "context"
 "errors"
```

```
 "log"

 "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"
)

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(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &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(ctx,
 &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 para 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.

```
/**
 * Connect a Lambda function to the PreSignUp trigger for a Cognito user pool
 * @param {{ region: string, userPoolId: string, handlerArn: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").UpdateUserPoolCommandOutput | null, unknown]>}
 */
```

```
export const addPreSignUpHandler = async ({
  region,
  userPoolId,
  handlerArn,
}) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({
      region,
    });

    const command = new UpdateUserPoolCommand({
      UserPoolId: userPoolId,
      LambdaConfig: {
        PreSignUp: handlerArn,
      },
    });

    const response = await cognitoClient.send(command);
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

- Para obtener más información sobre la API, consulta UpdateUserPoolla Referencia AWS SDK para JavaScript 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

`VerifySoftwareToken`Úselo con un AWS SDK o CLI

Los siguientes ejemplos de código muestran 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

SDK para .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 para .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 para C++ de la API.

CLI

AWS CLI

Confirmación del registro de una autenticación TOTP

En el siguiente ejemplo de `verify-software-token`, se completa el registro TOTP para el usuario actual.

```
aws cognito-idp verify-software-token \
    --access-token eyJra456defEXAMPLE \
    --user-code 123456
```

Salida:

```
{
    "Status": "SUCCESS"
}
```

Para obtener más información, consulte Adding MFA to a user pool en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla 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.

```
    // 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 para 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.fromEnvironment { 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).

SAP ABAP

SDK para SAP ABAP

> **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.

```
    TRY.
        DATA(lo_result) = lo_cgp->verifysoftwaretoken(
          iv_session = iv_session
          iv_usercode = iv_user_code
        ).

        ov_status = lo_result->get_status( ).
```

```
        IF ov_status = 'SUCCESS'.
          MESSAGE 'MFA token verified successfully.' TYPE 'I'.
        ELSE.
          MESSAGE |MFA verification status: { ov_status }.| TYPE 'I'.
        ENDIF.

      CATCH /aws1/cx_cgpcodemismatchex INTO DATA(lo_code_ex).
        MESSAGE 'Invalid MFA code provided.' TYPE 'E'.

      CATCH /aws1/cx_cgpenbsoftwaretokmf00 INTO DATA(lo_enabled_ex).
        MESSAGE 'Software token MFA is already enabled.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte VerifySoftwareTokenla referencia sobre la API ABAP del AWS SDK para SAP.

Swift

SDK para Swift

> **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 AWSClientRuntime
import AWSCognitoIdentityProvider

    /// Confirm that the user's TOTP authenticator is configured correctly by
    /// sending a code to it to check that it matches successfully.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongnitoIdentityProviderClient` to use.
    ///   - session: An authentication session previously returned by an
    ///     `associateSoftwareToken()` call.
    ///   - mfaCode: The 6-digit code currently displayed by the user's
    ///     authenticator, as provided by the user.
```

```
    func verifyTOTP(cipClient: CognitoIdentityProviderClient, session: String?,
mfaCode: String?) async {
        do {
            let output = try await cipClient.verifySoftwareToken(
                input: VerifySoftwareTokenInput(
                    session: session,
                    userCode: mfaCode
                )
            )

            guard let tokenStatus = output.status else {
                print("*** Unable to get the token's status.")
                return
            }
            print("=====> The token's status is: \(tokenStatus)")
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user has not been confirmed.")
            return
        } catch {
            print("*** Error verifying the MFA token!")
            return
        }
    }
```

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escenarios en los que Amazon Cognito Identity Provider utiliza AWS SDK

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes en Amazon Cognito Identity Provider con AWS SDK. Estos escenarios muestran cómo realizar tareas específicas llamando a varias funciones dentro del proveedor de identidades de Amazon Cognito o en combinación con otros Servicios de AWS. En cada escenario se incluye un enlace al código fuente completo, con instrucciones de configuración y ejecución del código.

Los escenarios requieren un nivel intermedio de experiencia para entender las acciones de servicio en su contexto.

Ejemplos

- Confirme automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un AWS SDK
- Migre automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un AWS SDK
- Obtenga tokens de cliente M2M y descubra las condiciones de Amazon Cognito mediante un AWS SDK
- Introducción a los grupos de usuarios
- Registrar un usuario con un grupo de usuarios de Amazon Cognito que requiera MFA mediante un AWS SDK
- Uso de los grupos de identidades y los flujos de identidades de Amazon Cognito
- Escriba datos de actividad personalizados con una función Lambda tras la autenticación de usuarios de Amazon Cognito mediante un AWS SDK

Confirme automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un AWS SDK

En los siguientes ejemplos 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.

```
import (
 "context"
 "errors"
 "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"
)

// 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(ctx context.Context, 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(
  ctx, 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(ctx context.Context, 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(ctx, 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(ctx, 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(ctx context.Context, 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(ctx, 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(ctx context.Context, stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup(ctx)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(ctx, stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]
 runner.helper.PopulateUserTable(ctx, stackOutputs["TableName"])

 runner.AddPreSignUpTrigger(ctx, stackOutputs["UserPoolId"],
 stackOutputs["AutoConfirmFunctionArn"])
 runner.resources.triggers = append(runner.resources.triggers, actions.PreSignUp)
 userName, password := runner.SignUpUser(ctx, stackOutputs["UserPoolClientId"],
 stackOutputs["TableName"])
 runner.helper.ListRecentLogEvents(ctx, stackOutputs["AutoConfirmFunction"])
 runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
  runner.SignInUser(ctx, stackOutputs["UserPoolClientId"], userName, password))

 runner.resources.Cleanup(ctx)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Controle el desencadenador `PreSignUp` con una función de Lambda.

```
import (
 "context"
 "log"
 "os"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 dynamodbtypes "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

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() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 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.

```
import (
 "context"
 "log"
 "strings"
 "time"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
 GetStackOutputs(ctx context.Context, stackName string) (actions.StackOutputs,
 error)
 PopulateUserTable(ctx context.Context, tableName string)
 GetKnownUsers(ctx context.Context, tableName string) (actions.UserList, error)
 AddKnownUser(ctx context.Context, tableName string, user actions.User)
 ListRecentLogEvents(ctx context.Context, 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(ctx context.Context, stackName
 string) (actions.StackOutputs, error) {
 return helper.cfnActor.GetOutputs(ctx, stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(ctx context.Context, 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(ctx, tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(ctx context.Context, tableName string)
 (actions.UserList, error) {
 knownUsers, err := helper.dynamoActor.Scan(ctx, 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(ctx context.Context, 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(ctx, 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(ctx context.Context,
 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(ctx, functionName)
 if err != nil {
  panic(err)
 }
```

```
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(ctx, 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

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(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &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(ctx,
 &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(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &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(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &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(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &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(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &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(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &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(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &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(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &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.

```
import (
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
```

```
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

// 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(ctx context.Context, 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(ctx, &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(ctx context.Context, tableName string) (UserList,
 error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(ctx, &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(ctx context.Context, 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(ctx, &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 abarque las acciones de CloudWatch Logs.

```
import (
 "context"
 "fmt"
 "log"

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

type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(ctx context.Context,
 functionName string) (types.LogStream, error) {
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.DescribeLogStreams(ctx,
 &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(ctx context.Context, 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(ctx,
 &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. CloudFormation

```
import (
 "context"
 "log"
```

```
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/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(ctx context.Context, stackName
 string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(ctx,
 &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.

```
import (
 "context"
 "log"
 "user_pools_and_lambda_triggers/actions"

 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)
```

```
// 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(ctx context.Context) {
 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(ctx, 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(ctx, 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 detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Go .
  - DeleteUser
  - InitiateAuth
  - SignUp
  - UpdateUserPool

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.

Configure una ejecución interactiva de “Escenario”. Los ejemplos JavaScript (v3) comparten un gestor de escenarios para simplificar los ejemplos complejos. El código fuente completo está activado GitHub.

```
import { AutoConfirm } from "./scenario-auto-confirm.js";

/**
 * The context is passed to every scenario. Scenario steps
 * will modify the context.
 */
const context = {
  errors: [],
  users: [
    {
      UserName: "test_user_1",
      UserEmail: "test_email_1@example.com",
    },
    {
      UserName: "test_user_2",
      UserEmail: "test_email_2@example.com",
    },
    {
      UserName: "test_user_3",
      UserEmail: "test_email_3@example.com",
    },
  ],
};

/**
 * Three Scenarios are created for the workflow. A Scenario is an orchestration
 class
 * that simplifies running a series of steps.
 */
export const scenarios = {
  // Demonstrate automatically confirming known users in a database.
  "auto-confirm": AutoConfirm(context),
};

// Call function if run directly
import { fileURLToPath } from "node:url";
import { parseScenarioArgs } from "@aws-doc-sdk-examples/lib/scenario/index.js";

if (process.argv[1] === fileURLToPath(import.meta.url)) {
  parseScenarioArgs(scenarios, {
    name: "Cognito user pools and triggers",
    description:
```

```
      "Demonstrate how to use the AWS SDKs to customize Amazon Cognito
 authentication behavior.",
  });
}
```

En este escenario, se muestra la confirmación automática de un usuario conocido. Orquesta los pasos del ejemplo.

```
import { wait } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import {
  Scenario,
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/scenario.js";

import {
  getStackOutputs,
  logCleanUpReminder,
  promptForStackName,
  promptForStackRegion,
  skipWhenErrors,
} from "./steps-common.js";
import { populateTable } from "./actions/dynamodb-actions.js";
import {
  addPreSignUpHandler,
  deleteUser,
  getUser,
  signIn,
  signUpUser,
} from "./actions/cognito-actions.js";
import {
  getLatestLogStreamForLambda,
  getLogEvents,
} from "./actions/cloudwatch-logs-actions.js";

/**
 * @typedef {{
 *   errors: Error[],
 *   password: string,
 *   users: { UserName: string, UserEmail: string }[],
 *   selectedUser?: string,
```

```
 *   stackName?: string,
 *   stackRegion?: string,
 *   token?: string,
 *   confirmDeleteSignedInUser?: boolean,
 *   TableName?: string,
 *   UserPoolClientId?: string,
 *   UserPoolId?: string,
 *   UserPoolArn?: string,
 *   AutoConfirmHandlerArn?: string,
 *   AutoConfirmHandlerName?: string
 * }} State
 */

const greeting = new ScenarioOutput(
  "greeting",
  (/** @type {State} */ state) => `This demo will populate some users into the \
database created as part of the "${state.stackName}" stack. \
Then the AutoConfirmHandler will be linked to the PreSignUp \
trigger from Cognito. Finally, you will choose a user to sign up.`,
  { skipWhen: skipWhenErrors },
);

const logPopulatingUsers = new ScenarioOutput(
  "logPopulatingUsers",
  "Populating the DynamoDB table with some users.",
  { skipWhenErrors: skipWhenErrors },
);

const logPopulatingUsersComplete = new ScenarioOutput(
  "logPopulatingUsersComplete",
  "Done populating users.",
  { skipWhen: skipWhenErrors },
);

const populateUsers = new ScenarioAction(
  "populateUsers",
  async (/** @type {State} */ state) => {
    const [_, err] = await populateTable({
      region: state.stackRegion,
      tableName: state.TableName,
      items: state.users,
    });
    if (err) {
      state.errors.push(err);
```

```
    }
  },
  {
    skipWhen: skipWhenErrors,
  },
);

const logSetupSignUpTrigger = new ScenarioOutput(
  "logSetupSignUpTrigger",
  "Setting up the PreSignUp trigger for the Cognito User Pool.",
  { skipWhen: skipWhenErrors },
);

const setupSignUpTrigger = new ScenarioAction(
  "setupSignUpTrigger",
  async (/** @type {State} */ state) => {
    const [_, err] = await addPreSignUpHandler({
      region: state.stackRegion,
      userPoolId: state.UserPoolId,
      handlerArn: state.AutoConfirmHandlerArn,
    });
    if (err) {
      state.errors.push(err);
    }
  },
  {
    skipWhen: skipWhenErrors,
  },
);

const logSetupSignUpTriggerComplete = new ScenarioOutput(
  "logSetupSignUpTriggerComplete",
  (
    /** @type {State} */ state,
  ) => `The lambda function "${state.AutoConfirmHandlerName}" \
has been configured as the PreSignUp trigger handler for the user pool
 "${state.UserPoolId}".`,
  { skipWhen: skipWhenErrors },
);

const selectUser = new ScenarioInput(
  "selectedUser",
  "Select a user to sign up.",
  {
```

```
    type: "select",
    choices: (/** @type {State} */ state) => state.users.map((u) => u.UserName),
    skipWhen: skipWhenErrors,
    default: (/** @type {State} */ state) => state.users[0].UserName,
  },
);

const checkIfUserAlreadyExists = new ScenarioAction(
  "checkIfUserAlreadyExists",
  async (/** @type {State} */ state) => {
    const [user, err] = await getUser({
      region: state.stackRegion,
      userPoolId: state.UserPoolId,
      username: state.selectedUser,
    });

    if (err?.name === "UserNotFoundException") {
      // Do nothing. We're not expecting the user to exist before
      // sign up is complete.
      return;
    }

    if (err) {
      state.errors.push(err);
      return;
    }

    if (user) {
      state.errors.push(
        new Error(
          `The user "${state.selectedUser}" already exists in the user pool
 "${state.UserPoolId}".`,
        ),
      );
    }
  },
  {
    skipWhen: skipWhenErrors,
  },
);

const createPassword = new ScenarioInput(
  "password",
```

```
  "Enter a password that has at least eight characters, uppercase, lowercase,
 numbers and symbols.",
  { type: "password", skipWhen: skipWhenErrors, default: "Abcd1234!" },
);

const logSignUpExistingUser = new ScenarioOutput(
  "logSignUpExistingUser",
  (/** @type {State} */ state) => `Signing up user "${state.selectedUser}".`,
  { skipWhen: skipWhenErrors },
);

const signUpExistingUser = new ScenarioAction(
  "signUpExistingUser",
  async (/** @type {State} */ state) => {
    const signUp = (password) =>
      signUpUser({
        region: state.stackRegion,
        userPoolClientId: state.UserPoolClientId,
        username: state.selectedUser,
        email: state.users.find((u) => u.UserName === state.selectedUser)
          .UserEmail,
        password,
      });

    let [_, err] = await signUp(state.password);

    while (err?.name === "InvalidPasswordException") {
      console.warn("The password you entered was invalid.");
      await createPassword.handle(state);
      [_, err] = await signUp(state.password);
    }

    if (err) {
      state.errors.push(err);
    }
  },
  { skipWhen: skipWhenErrors },
);

const logSignUpExistingUserComplete = new ScenarioOutput(
  "logSignUpExistingUserComplete",
  (/** @type {State} */ state) =>
    `"${state.selectedUser} was signed up successfully.`,
  { skipWhen: skipWhenErrors },
```

```
);

const logLambdaLogs = new ScenarioAction(
  "logLambdaLogs",
  async (/** @type {State} */ state) => {
    console.log(
      "Waiting a few seconds to let Lambda write to CloudWatch Logs...\n",
    );
    await wait(10);

    const [logStream, logStreamErr] = await getLatestLogStreamForLambda({
      functionName: state.AutoConfirmHandlerName,
      region: state.stackRegion,
    });
    if (logStreamErr) {
      state.errors.push(logStreamErr);
      return;
    }

    console.log(
      `Getting some recent events from log stream "${logStream.logStreamName}"`,
    );
    const [logEvents, logEventsErr] = await getLogEvents({
      functionName: state.AutoConfirmHandlerName,
      region: state.stackRegion,
      eventCount: 10,
      logStreamName: logStream.logStreamName,
    });
    if (logEventsErr) {
      state.errors.push(logEventsErr);
      return;
    }

    console.log(logEvents.map((ev) => `\t${ev.message}`).join(""));
  },
  { skipWhen: skipWhenErrors },
);

const logSignInUser = new ScenarioOutput(
  "logSignInUser",
  (/** @type {State} */ state) => `Let's sign in as ${state.selectedUser}`,
  { skipWhen: skipWhenErrors },
);
```

```
const signInUser = new ScenarioAction(
  "signInUser",
  async (/** @type {State} */ state) => {
    const [response, err] = await signIn({
      region: state.stackRegion,
      clientId: state.UserPoolClientId,
      username: state.selectedUser,
      password: state.password,
    });

    if (err?.name === "PasswordResetRequiredException") {
      state.errors.push(new Error("Please reset your password."));
      return;
    }

    if (err) {
      state.errors.push(err);
      return;
    }

    state.token = response?.AuthenticationResult?.AccessToken;
  },
  { skipWhen: skipWhenErrors },
);

const logSignInUserComplete = new ScenarioOutput(
  "logSignInUserComplete",
  (/** @type {State} */ state) =>
    `Successfully signed in. Your access token starts with:
 ${state.token.slice(0, 11)}`,
  { skipWhen: skipWhenErrors },
);

const confirmDeleteSignedInUser = new ScenarioInput(
  "confirmDeleteSignedInUser",
  "Do you want to delete the currently signed in user?",
  { type: "confirm", skipWhen: skipWhenErrors },
);

const deleteSignedInUser = new ScenarioAction(
  "deleteSignedInUser",
  async (/** @type {State} */ state) => {
    const [_, err] = await deleteUser({
      region: state.stackRegion,
```

```
      accessToken: state.token,
    });

    if (err) {
      state.errors.push(err);
    }
  },
  {
    skipWhen: (/** @type {State} */ state) =>
      skipWhenErrors(state) || !state.confirmDeleteSignedInUser,
  },
);

const logErrors = new ScenarioOutput(
  "logErrors",
  (/** @type {State}*/ state) => {
    const errorList = state.errors
      .map((err) => ` - ${err.name}: ${err.message}`)
      .join("\n");
    return `Scenario errors found:\n${errorList}`;
  },
  {
    // Don't log errors when there aren't any!
    skipWhen: (/** @type {State} */ state) => state.errors.length === 0,
  },
);

export const AutoConfirm = (context) =>
  new Scenario(
    "AutoConfirm",
    [
      promptForStackName,
      promptForStackRegion,
      getStackOutputs,
      greeting,
      logPopulatingUsers,
      populateUsers,
      logPopulatingUsersComplete,
      logSetupSignUpTrigger,
      setupSignUpTrigger,
      logSetupSignUpTriggerComplete,
      selectUser,
      checkIfUserAlreadyExists,
      createPassword,
```

```
      logSignUpExistingUser,
      signUpExistingUser,
      logSignUpExistingUserComplete,
      logLambdaLogs,
      logSignInUser,
      signInUser,
      logSignInUserComplete,
      confirmDeleteSignedInUser,
      deleteSignedInUser,
      logCleanUpReminder,
      logErrors,
    ],
    context,
  );
```

Estos son pasos que se comparten con otros escenarios.

```
import {
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/scenario.js";
import { getCfnOutputs } from "@aws-doc-sdk-examples/lib/sdk/cfn-outputs.js";

export const skipWhenErrors = (state) => state.errors.length > 0;

export const getStackOutputs = new ScenarioAction(
  "getStackOutputs",
  async (state) => {
    if (!state.stackName || !state.stackRegion) {
      state.errors.push(
        new Error(
          "No stack name or region provided. The stack name and \
region are required to fetch CFN outputs relevant to this example.",
        ),
      );
      return;
    }

    const outputs = await getCfnOutputs(state.stackName, state.stackRegion);
    Object.assign(state, outputs);
  },
```

```
);

export const promptForStackName = new ScenarioInput(
  "stackName",
  "Enter the name of the stack you deployed earlier.",
  { type: "input", default: "PoolsAndTriggersStack" },
);

export const promptForStackRegion = new ScenarioInput(
  "stackRegion",
  "Enter the region of the stack you deployed earlier.",
  { type: "input", default: "us-east-1" },
);

export const logCleanUpReminder = new ScenarioOutput(
  "logCleanUpReminder",
  "All done. Remember to run 'cdk destroy' to teardown the stack.",
  { skipWhen: skipWhenErrors },
);
```

Un controlador para el desencadenador `PreSignUp` con una función de Lambda.

```
import type { PreSignUpTriggerEvent, Handler } from "aws-lambda";
import type { UserRepository } from "./user-repository";
import { DynamoDBUserRepository } from "./user-repository";

export class PreSignUpHandler {
  private userRepository: UserRepository;

  constructor(userRepository: UserRepository) {
    this.userRepository = userRepository;
  }

  private isPreSignUpTriggerSource(event: PreSignUpTriggerEvent): boolean {
    return event.triggerSource === "PreSignUp_SignUp";
  }

  private getEventUserEmail(event: PreSignUpTriggerEvent): string {
    return event.request.userAttributes.email;
  }

  async handlePreSignUpTriggerEvent(
```

```
    event: PreSignUpTriggerEvent,
  ): Promise<PreSignUpTriggerEvent> {
    console.log(
      `Received presignup from ${event.triggerSource} for user
 '${event.userName}'`,
    );

    if (!this.isPreSignUpTriggerSource(event)) {
      return event;
    }

    const eventEmail = this.getEventUserEmail(event);
    console.log(`Looking up email ${eventEmail}.`);
    const storedUserInfo =
      await this.userRepository.getUserInfoByEmail(eventEmail);

    if (!storedUserInfo) {
      console.log(
        `Email ${eventEmail} not found. Email verification is required.`,
      );
      return event;
    }

    if (storedUserInfo.UserName !== event.userName) {
      console.log(
        `UserEmail ${eventEmail} found, but stored UserName
 '${storedUserInfo.UserName}' does not match supplied UserName
 '${event.userName}'. Verification is required.`,
      );
    } else {
      console.log(
        `UserEmail ${eventEmail} found with matching UserName
 ${storedUserInfo.UserName}. User is confirmed.`,
      );
      event.response.autoConfirmUser = true;
      event.response.autoVerifyEmail = true;
    }
    return event;
  }
}

const createPreSignUpHandler = (): PreSignUpHandler => {
  const tableName = process.env.TABLE_NAME;
  if (!tableName) {
```

```
    throw new Error("TABLE_NAME environment variable is not set");
  }

  const userRepository = new DynamoDBUserRepository(tableName);
  return new PreSignUpHandler(userRepository);
};

export const handler: Handler = async (event: PreSignUpTriggerEvent) => {
  const preSignUpHandler = createPreSignUpHandler();
  return preSignUpHandler.handlePreSignUpTriggerEvent(event);
};
```

Acciones del módulo de CloudWatch registros.

```
import {
  CloudWatchLogsClient,
  GetLogEventsCommand,
  OrderBy,
  paginateDescribeLogStreams,
} from "@aws-sdk/client-cloudwatch-logs";

/**
 * Get the latest log stream for a Lambda function.
 * @param {{ functionName: string, region: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cloudwatch-logs").LogStream | null,
 unknown]>}
 */
export const getLatestLogStreamForLambda = async ({ functionName, region }) => {
  try {
    const logGroupName = `/aws/lambda/${functionName}`;
    const cwlClient = new CloudWatchLogsClient({ region });
    const paginator = paginateDescribeLogStreams(
      { client: cwlClient },
      {
        descending: true,
        limit: 1,
        orderBy: OrderBy.LastEventTime,
        logGroupName,
      },
    );
```

```
    for await (const page of paginator) {
      return [page.logStreams[0], null];
    }
  } catch (err) {
    return [null, err];
  }
};

/**
 * Get the log events for a Lambda function's log stream.
 * @param {{
 *   functionName: string,
 *   logStreamName: string,
 *   eventCount: number,
 *   region: string
 * }} config
 * @returns {Promise<[import("@aws-sdk/client-cloudwatch-logs").OutputLogEvent[]
 | null, unknown]>}
 */
export const getLogEvents = async ({
  functionName,
  logStreamName,
  eventCount,
  region,
}) => {
  try {
    const cwlClient = new CloudWatchLogsClient({ region });
    const logGroupName = `/aws/lambda/${functionName}`;
    const response = await cwlClient.send(
      new GetLogEventsCommand({
        logStreamName: logStreamName,
        limit: eventCount,
        logGroupName: logGroupName,
      }),
    );

    return [response.events, null];
  } catch (err) {
    return [null, err];
  }
};
```

Módulo de acciones de Amazon Cognito.

```
import {
  AdminGetUserCommand,
  CognitoIdentityProviderClient,
  DeleteUserCommand,
  InitiateAuthCommand,
  SignUpCommand,
  UpdateUserPoolCommand,
} from "@aws-sdk/client-cognito-identity-provider";

/**
 * Connect a Lambda function to the PreSignUp trigger for a Cognito user pool
 * @param {{ region: string, userPoolId: string, handlerArn: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").UpdateUserPoolCommandOutput | null, unknown]>}
 */
export const addPreSignUpHandler = async ({
  region,
  userPoolId,
  handlerArn,
}) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({
      region,
    });

    const command = new UpdateUserPoolCommand({
      UserPoolId: userPoolId,
      LambdaConfig: {
        PreSignUp: handlerArn,
      },
    });

    const response = await cognitoClient.send(command);
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Attempt to register a user to a user pool with a given username and password.
```

```
 * @param {{
 *   region: string,
 *   userPoolClientId: string,
 *   username: string,
 *   email: string,
 *   password: string
 * }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").SignUpCommandOutput | null, unknown]>}
 */
export const signUpUser = async ({
  region,
  userPoolClientId,
  username,
  email,
  password,
}) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({
      region,
    });

    const response = await cognitoClient.send(
      new SignUpCommand({
        ClientId: userPoolClientId,
        Username: username,
        Password: password,
        UserAttributes: [{ Name: "email", Value: email }],
      }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Sign in a user to Amazon Cognito using a username and password authentication
 flow.
 * @param {{ region: string, clientId: string, username: string, password:
 string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").InitiateAuthCommandOutput | null, unknown]>}
 */
```

```
export const signIn = async ({ region, clientId, username, password }) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({ region });
    const response = await cognitoClient.send(
      new InitiateAuthCommand({
        AuthFlow: "USER_PASSWORD_AUTH",
        ClientId: clientId,
        AuthParameters: { USERNAME: username, PASSWORD: password },
      }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Retrieve an existing user from a user pool.
 * @param {{ region: string, userPoolId: string, username: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").AdminGetUserCommandOutput | null, unknown]>}
 */
export const getUser = async ({ region, userPoolId, username }) => {
  try {
    const cognitoClient = new CognitoIdentityProviderClient({ region });
    const response = await cognitoClient.send(
      new AdminGetUserCommand({
        UserPoolId: userPoolId,
        Username: username,
      }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};

/**
 * Delete the signed-in user. Useful for allowing a user to delete their
 * own profile.
 * @param {{ region: string, accessToken: string }} config
 * @returns {Promise<[import("@aws-sdk/client-cognito-identity-
provider").DeleteUserCommandOutput | null, unknown]>}
 */
```

```
export const deleteUser = async ({ region, accessToken }) => {
  try {
    const client = new CognitoIdentityProviderClient({ region });
    const response = await client.send(
      new DeleteUserCommand({ AccessToken: accessToken }),
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

Módulo de acciones de DynamoDB.

```
import { DynamoDBClient } from "@aws-sdk/client-dynamodb";
import {
  BatchWriteCommand,
  DynamoDBDocumentClient,
} from "@aws-sdk/lib-dynamodb";

/**
 * Populate a DynamoDB table with provide items.
 * @param {{ region: string, tableName: string, items: Record<string,
 unknown>[] }} config
 * @returns {Promise<[import("@aws-sdk/lib-dynamodb").BatchWriteCommandOutput |
 null, unknown]>}
 */
export const populateTable = async ({ region, tableName, items }) => {
  try {
    const ddbClient = new DynamoDBClient({ region });
    const docClient = DynamoDBDocumentClient.from(ddbClient);
    const response = await docClient.send(
      new BatchWriteCommand({
        RequestItems: {
          [tableName]: items.map((item) => ({
            PutRequest: {
              Item: item,
            },
          })),
        },
      }),
```

```
    );
    return [response, null];
  } catch (err) {
    return [null, err];
  }
};
```

- Para obtener detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para JavaScript .
  - DeleteUser
  - InitiateAuth
  - SignUp
  - UpdateUserPool

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo empezar 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 AWS SDK

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 (
 "context"
 "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(ctx context.Context, 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(
  ctx, 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(ctx context.Context, 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(ctx, 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(ctx, 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(ctx context.Context, 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(ctx, 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(ctx, 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(ctx, 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(ctx context.Context, stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup(ctx)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(ctx, stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]

 runner.AddMigrateUserTrigger(ctx, stackOutputs["UserPoolId"],
 stackOutputs["MigrateUserFunctionArn"])
 runner.resources.triggers = append(runner.resources.triggers,
 actions.UserMigration)
 resetNeeded, user := runner.SignInUser(ctx, stackOutputs["TableName"],
 stackOutputs["UserPoolClientId"])
 if resetNeeded {
  runner.helper.ListRecentLogEvents(ctx, stackOutputs["MigrateUserFunction"])
  runner.ResetPassword(ctx, stackOutputs["UserPoolClientId"], user)
 }

 runner.resources.Cleanup(ctx)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Controle el desencadenador `MigrateUser` con una función de Lambda.

```
import (
 "context"
 "log"
 "os"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/expression"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
)

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 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() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 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.

```
import (
 "context"
 "log"
 "strings"
 "time"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
```

```
 GetStackOutputs(ctx context.Context, stackName string) (actions.StackOutputs,
 error)
 PopulateUserTable(ctx context.Context, tableName string)
 GetKnownUsers(ctx context.Context, tableName string) (actions.UserList, error)
 AddKnownUser(ctx context.Context, tableName string, user actions.User)
 ListRecentLogEvents(ctx context.Context, 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(ctx context.Context, stackName
 string) (actions.StackOutputs, error) {
```

```
 return helper.cfnActor.GetOutputs(ctx, stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(ctx context.Context, 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(ctx, tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(ctx context.Context, tableName string)
 (actions.UserList, error) {
 knownUsers, err := helper.dynamoActor.Scan(ctx, 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(ctx context.Context, 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(ctx, 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(ctx context.Context,
 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(ctx, functionName)
 if err != nil {
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(ctx, 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

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(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &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(ctx,
 &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(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &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(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &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(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &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(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &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(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &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(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &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(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &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.

```
import (
```

```
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

// 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(ctx context.Context, 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(ctx, &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(ctx context.Context, tableName string) (UserList,
 error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(ctx, &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(ctx context.Context, 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(ctx, &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 abarque las acciones de CloudWatch Logs.

```
import (
 "context"
 "fmt"
 "log"

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

type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(ctx context.Context,
 functionName string) (types.LogStream, error) {
```

```
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.DescribeLogStreams(ctx,
 &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(ctx context.Context, 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(ctx,
 &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. CloudFormation

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/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(ctx context.Context, stackName
 string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(ctx,
 &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.

```
import (
 "context"
 "log"
```

```
 "user_pools_and_lambda_triggers/actions"

 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// 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(ctx context.Context) {
 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(ctx, 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(ctx, 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 detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

Obtenga tokens de cliente M2M y descubra las condiciones de Amazon Cognito mediante un AWS SDK

El siguiente ejemplo de código muestra cómo utilizar la autorización de máquina a máquina (M2M) de Amazon Cognito y la detección de términos.

- Implemente una CloudFormation pila con los recursos de Cognito (grupo de usuarios, servidor de recursos, cliente de aplicaciones, documentos de condiciones).
- Recupera el secreto del cliente de la aplicación con DescribeUserPoolClient.
- Obtenga los tokens de acceso M2M con GetClientToken.
- Descubra los documentos de Términos con DescribeTermsByClient.
- Limpie eliminando la CloudFormation pila.

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.

Ejecutar un escenario interactivo en un símbolo del sistema.

```
class ClientTokenAndTermsScenario:
    """
    Runs an interactive scenario that demonstrates the Amazon Cognito
    GetClientToken and DescribeTermsByClient APIs.
    """

    def __init__(
        self,
        cognito_wrapper: CognitoIdentityProviderWrapper,
        cfn_client: BaseClient,
    ):
        """
        :param cognito_wrapper: An instance of the CognitoIdentityProviderWrapper
 class.
```

```
        :param cfn_client: A Boto3 CloudFormation client.
        """
        self.cognito_wrapper = cognito_wrapper
        self.cfn_client = cfn_client
        self.stack_name = None
        self.user_pool_id = None
        self.client_id = None
        self.client_secret = None

    def run(self) -> None:
        """Runs the scenario from start to finish."""
        print(SEPARATOR)
        print("Welcome to the Amazon Cognito Client Token and Terms Scenario.")
        print(SEPARATOR)
        print("This example creates Cognito resources in a CloudFormation stack,
then")
        print("demonstrates the new GetClientToken and DescribeTermsByClient
APIs.")
        print()

        try:
            self.setup()
            self.get_client_secret()
            self.demonstrate_get_client_token()
            self.demonstrate_describe_terms()
        except Exception:
            logger.exception("An error occurred during the scenario.")
        finally:
            self.cleanup()

        print(SEPARATOR)
        print("Amazon Cognito Client Token and Terms scenario completed.")

    # ------------------------------------------------------------------
    # Phase 1: Setup - deploy CloudFormation stack
    # ------------------------------------------------------------------
    def setup(self) -> None:
        """Deploys the CloudFormation prerequisite stack and retrieves
outputs."""
        stack_name = q.ask(
            f"\nEnter a name for the CloudFormation stack [{DEFAULT_STACK_NAME}]:
"
        )
        if not stack_name:
```

```
            stack_name = DEFAULT_STACK_NAME

        # Check whether the stack name already exists.
        while self._stack_exists(stack_name):
            print(f"A stack named '{stack_name}' already exists.")
            stack_name = q.ask("Enter a different stack name: ")

        self.stack_name = stack_name

        template_body = self._read_template()
        print(f"Deploying CloudFormation stack: {self.stack_name}")
        self.cfn_client.create_stack(
            StackName=self.stack_name,
            TemplateBody=template_body,
            Capabilities=["CAPABILITY_IAM"],
        )
        print(f"CloudFormation stack creation started: {self.stack_name}")
        print("Waiting for CloudFormation stack creation to complete...")

        waiter = self.cfn_client.get_waiter("stack_create_complete")
        try:
            waiter.wait(
                StackName=self.stack_name,
                WaiterConfig={"Delay": 10, "MaxAttempts": 60},
            )
        except WaiterError:
            logger.error("Stack creation failed or timed out.")
            self._print_stack_events()
            raise

        print("CloudFormation stack creation complete.")
        self._retrieve_stack_outputs()
        print(SEPARATOR)

    def _stack_exists(self, stack_name: str) -> bool:
        """Returns True if a stack with the given name exists and is not
deleted."""
        if not stack_name:
            return False
        try:
            response = self.cfn_client.describe_stacks(StackName=stack_name)
            stacks = response.get("Stacks", list())
            for stack in stacks:
                status = stack.get("StackStatus", "")
```

```
                if "DELETE_COMPLETE" not in status:
                    return True
            return False
        except ClientError as err:
            if "does not exist" in str(err):
                return False
            raise

    def _read_template(self) -> str:
        """Reads the CloudFormation template from the local file system."""
        template_path = os.path.join(
            os.path.dirname(os.path.abspath(__file__)), CFN_TEMPLATE_PATH
        )
        if not os.path.exists(template_path):
            raise FileNotFoundError(
                f"CloudFormation template not found at {template_path}. "
                "Ensure the resources/cfn_template.yaml file exists."
            )
        with open(template_path, "r") as f:
            return f.read()

    def _retrieve_stack_outputs(self) -> None:
        """Retrieves and displays outputs from the CloudFormation stack."""
        response = self.cfn_client.describe_stacks(StackName=self.stack_name)
        stacks = response.get("Stacks", list())
        if not stacks:
            raise RuntimeError(f"Stack '{self.stack_name}' not found.")

        outputs = stacks[0].get("Outputs", list())
        for output in outputs:
            key = output.get("OutputKey", "")
            value = output.get("OutputValue", "")
            print(f"Stack output {key}: {value}")
            if key == "UserPoolId":
                self.user_pool_id = value
            elif key == "ClientId":
                self.client_id = value

        if not self.user_pool_id or not self.client_id:
            raise RuntimeError(
                "Failed to retrieve UserPoolId or ClientId from stack outputs."
            )

    def _print_stack_events(self) -> None:
```

```
        """Prints recent stack events to help diagnose failures."""
        try:
            response =
self.cfn_client.describe_stack_events(StackName=self.stack_name)
            events = response.get("StackEvents", list())
            for event in events[:10]:
                status = event.get("ResourceStatus", "")
                reason = event.get("ResourceStatusReason", "")
                resource = event.get("LogicalResourceId", "")
                if "FAILED" in status:
                    print(f"  FAILED: {resource} - {reason}")
        except ClientError:
            logger.warning("Could not retrieve stack events.")

    # ------------------------------------------------------------------
    # Phase 2: Retrieve the app client secret
    # ------------------------------------------------------------------
    def get_client_secret(self) -> None:
        """Calls DescribeUserPoolClient to retrieve the client secret."""
        print(SEPARATOR)
        print("Retrieving the app client secret...")
        client_info = self.cognito_wrapper.describe_user_pool_client(
            self.user_pool_id, self.client_id
        )
        self.client_secret = client_info.get("ClientSecret", "")
        if not self.client_secret:
            raise RuntimeError(
                "Client secret not found. Ensure the app client was created "
                "with GenerateSecret: true."
            )
        print("Client secret retrieved successfully.")
        print(SEPARATOR)

    # ------------------------------------------------------------------
    # Phase 3: Demonstrate GetClientToken
    # ------------------------------------------------------------------
    def demonstrate_get_client_token(self) -> None:
        """Demonstrates obtaining M2M access tokens with GetClientToken."""
        print(SEPARATOR)
        print("Obtaining an M2M access token with GetClientToken...")
        print()

        # 3a: Get token with all scopes (no scopes param = all configured scopes)
        print("Step 1: Request a token with all configured scopes.")
```

```
        auth_result = self.cognito_wrapper.get_client_token(
            client_id=self.client_id,
            client_secret=self.client_secret,
        )
        self._display_token_result(auth_result, "all scopes")

        q.ask("\nPress Enter to continue...")

        # 3b: Get token with specific scopes
        print()
        print("Step 2: Request a token with a specific scope (my-m2m-api/read).")
        scoped_result = self.cognito_wrapper.get_client_token(
            client_id=self.client_id,
            client_secret=self.client_secret,
            scopes=["my-m2m-api/read"],
        )
        self._display_token_result(scoped_result, "my-m2m-api/read")

        print()
        print("Token was successfully scoped to the requested permissions.")
        print(SEPARATOR)

    @staticmethod
    def _display_token_result(auth_result: dict, scope_description: str) -> None:
        """Displays token metadata."""
        token_type = auth_result.get("TokenType", "Unknown")
        expires_in = auth_result.get("ExpiresIn", 0)
        access_token = auth_result.get("AccessToken", "")
        preview = access_token[:20] + "..." if len(access_token) > 20 else
access_token
        print(f"  Scope:      {scope_description}")
        print(f"  Token type: {token_type}")
        print(f"  Expires in: {expires_in} seconds")
        print(f"  Token preview: {preview}")

    # ------------------------------------------------------------------
    # Phase 4: Demonstrate DescribeTermsByClient
    # ------------------------------------------------------------------
    def demonstrate_describe_terms(self) -> None:
        """Demonstrates discovering Terms documents with
DescribeTermsByClient."""
        print(SEPARATOR)
        print("Discovering Terms documents with DescribeTermsByClient...")
        print()
```

```
        # 4a: Look up terms-of-use
        print("Looking up terms-of-use for the app client...")
        terms_of_use = self.cognito_wrapper.describe_terms_by_client(
            user_pool_id=self.user_pool_id,
            client_id=self.client_id,
            terms_name="terms-of-use",
        )
        self._display_terms(terms_of_use)

        q.ask("\nPress Enter to continue...")
        print()

        # 4b: Look up privacy-policy
        print("Looking up privacy-policy for the app client...")
        privacy_policy = self.cognito_wrapper.describe_terms_by_client(
            user_pool_id=self.user_pool_id,
            client_id=self.client_id,
            terms_name="privacy-policy",
        )
        self._display_terms(privacy_policy)

        print()
        print(
            "DescribeTermsByClient lets you discover which Terms documents are "
            "associated with a client without knowing the Terms resource ID up
front."
        )
        print(SEPARATOR)

    @staticmethod
    def _display_terms(terms: dict) -> None:
        """Displays a Terms document's details."""
        print(f"  Terms ID:          {terms.get('TermsId', 'N/A')}")
        print(f"  Terms name:        {terms.get('TermsName', 'N/A')}")
        print(f"  Client ID:         {terms.get('ClientId', 'N/A')}")
        print(f"  User pool ID:      {terms.get('UserPoolId', 'N/A')}")
        print(f"  Enforcement:       {terms.get('Enforcement', 'N/A')}")
        print(f"  Terms source:      {terms.get('TermsSource', 'N/A')}")
        links = terms.get("Links", dict())
        if links:
            print("  Links:")
            for lang_tag, url in links.items():
                print(f"    {lang_tag}: {url}")
```

```
        # Cognito returns CreationDate / LastModifiedDate as datetime objects
        # (boto3 deserializes the timestamps), so display them directly.
        creation_date = terms.get("CreationDate", None)
        if creation_date is not None:
            print(f"  Created:           {creation_date}")
        modified_date = terms.get("LastModifiedDate", None)
        if modified_date is not None:
            print(f"  Last modified:     {modified_date}")

    # ------------------------------------------------------------------
    # Phase 5: Cleanup - delete CloudFormation stack
    # ------------------------------------------------------------------
    def cleanup(self) -> None:
        """Prompts the user and deletes the CloudFormation stack."""
        if self.stack_name is None:
            return

        print(SEPARATOR)
        answer = q.ask(
            "Do you want to delete the CloudFormation stack and all resources?
(y/n) "
        )
        if answer.lower() != "y":
            print(
                f"Resources will remain. You can delete the stack later "
                f"through the AWS Console or CLI."
            )
            print(f"Stack name: {self.stack_name}")
            return

        print(
            f"CloudFormation stack '{self.stack_name}' is being deleted. "
            f"This may take a few minutes."
        )
        try:
            self.cfn_client.delete_stack(StackName=self.stack_name)
            waiter = self.cfn_client.get_waiter("stack_delete_complete")
            print("Waiting for CloudFormation stack deletion to complete...")
            waiter.wait(
                StackName=self.stack_name,
                WaiterConfig={"Delay": 10, "MaxAttempts": 60},
            )
            print(f"CloudFormation stack '{self.stack_name}' has been deleted.")
            print("All Cognito resources have been removed.")
```

```
        except (ClientError, WaiterError):
            logger.warning(
                "Stack deletion encountered an error. Attempting force delete."
            )
            try:
                self.cfn_client.delete_stack(
                    StackName=self.stack_name,
                    DeletionMode="FORCE_DELETE_STACK",
                )
                print("Force delete initiated. Please verify in the console.")
            except ClientError:
                logger.error(
                    "Force delete also failed. Clean up manually via the
 console."
                )
```

Cree una clase que agrupe las operaciones del proveedor de identidad de Amazon Cognito.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito Identity Provider actions."""

    def __init__(self, cognito_idp_client: BaseClient):
        """
        Initializes the CognitoIdentityProviderWrapper with an Amazon Cognito
        Identity Provider client.

        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        """
        self.cognito_idp_client = cognito_idp_client
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK para Python (Boto3).
  - DescribeTermsByClient
  - DescribeUserPoolClient
  - GetClientToken

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

Introducción a los grupos de usuarios

En el siguiente ejemplo de código, se muestra cómo:

- Creación de un grupo de usuarios
- Creación de un cliente de aplicación
- Configuración de un dominio para el grupo de usuarios
- Creación de un usuario como administrador
- Habilitación del autorregistro
- Muestra de usuarios del grupo de usuarios
- Eliminar recursos

Bash

AWS CLI con el script Bash

> **Note**
>
> Hay más información. GitHub Encuentre el ejemplo completo y obtenga información sobre cómo configurarlo y ejecutarlo en el repositorio de Tutoriales para desarrolladores de ejemplo.

```
#!/bin/bash

# Amazon Cognito User Pools Getting Started Script
# This script creates and configures an Amazon Cognito user pool with an app
 client

set -euo pipefail

# Security: Set restrictive umask
umask 0077

# Set up logging with secure permissions
```

```
LOG_FILE="cognito-user-pool-setup.log"
touch "$LOG_FILE"
chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Cognito User Pool setup script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to check for errors in command output
check_error() {
  local output=$1
  local cmd=$2

  if echo "$output" | grep -qi "error\|failed"; then
    echo "ERROR: Command failed: $cmd" >&2
    echo "Output: $output" >&2
    cleanup_on_error
    exit 1
  fi
}

# Function to check AWS CLI return code
check_aws_error() {
  local exit_code=$?
  local cmd=$1

  if [ $exit_code -ne 0 ]; then
    echo "ERROR: AWS CLI command failed with exit code $exit_code: $cmd" >&2
    cleanup_on_error
    exit "$exit_code"
  fi
}

# Function to clean up resources on error
cleanup_on_error() {
  echo "Error encountered. Attempting to clean up resources..." >&2

  if [ -n "${DOMAIN_NAME:-}" ] && [ -n "${USER_POOL_ID:-}" ]; then
    echo "Deleting user pool domain: $DOMAIN_NAME" >&2
    aws cognito-idp delete-user-pool-domain \
      --user-pool-id "$USER_POOL_ID" \
      --domain "$DOMAIN_NAME" 2>/dev/null || true
  fi
```

```
  if [ -n "${USER_POOL_ID:-}" ]; then
    echo "Deleting user pool: $USER_POOL_ID" >&2
    aws cognito-idp delete-user-pool \
      --user-pool-id "$USER_POOL_ID" 2>/dev/null || true
  fi
}

# Set trap for cleanup on exit
trap cleanup_on_error EXIT ERR

# Validate AWS CLI is installed and configured
if ! command -v aws &> /dev/null; then
  echo "ERROR: AWS CLI is not installed" >&2
  exit 1
fi

# Validate jq is installed
if ! command -v jq &> /dev/null; then
  echo "ERROR: jq is not installed" >&2
  exit 1
fi

# Validate openssl is installed
if ! command -v openssl &> /dev/null; then
  echo "ERROR: openssl is not installed" >&2
  exit 1
fi

if ! aws sts get-caller-identity &> /dev/null; then
  echo "ERROR: AWS CLI is not configured or credentials are invalid" >&2
  exit 1
fi

# Get the current AWS region
AWS_REGION=$(aws configure get region)
if [ -z "$AWS_REGION" ]; then
  AWS_REGION="us-east-1" # Default region if not configured
fi
echo "Using AWS Region: $AWS_REGION"

# Validate region format
if ! [[ "$AWS_REGION" =~ ^[a-z]{2}-[a-z]+-[0-9]{1}$ ]]; then
  echo "ERROR: Invalid AWS region format: $AWS_REGION" >&2
  exit 1
```

```
fi

# Generate random identifier for resource names using secure method
RANDOM_ID=$(openssl rand -hex 6)
if [ -z "$RANDOM_ID" ]; then
  echo "ERROR: Failed to generate random identifier" >&2
  exit 1
fi

USER_POOL_NAME="MyUserPool-${RANDOM_ID}"
APP_CLIENT_NAME="MyAppClient-${RANDOM_ID}"
DOMAIN_NAME="my-auth-domain-${RANDOM_ID}"

# Validate resource names don't exceed limits
if [ ${#USER_POOL_NAME} -gt 128 ]; then
  echo "ERROR: User pool name exceeds maximum length of 128 characters" >&2
  exit 1
fi

if [ ${#APP_CLIENT_NAME} -gt 128 ]; then
  echo "ERROR: App client name exceeds maximum length of 128 characters" >&2
  exit 1
fi

echo "Using random identifier: $RANDOM_ID"
echo "User pool name: $USER_POOL_NAME"
echo "App client name: $APP_CLIENT_NAME"
echo "Domain name: $DOMAIN_NAME"

# Step 1: Create a User Pool
echo "Creating user pool..."
USER_POOL_OUTPUT=$(aws cognito-idp create-user-pool \
  --pool-name "$USER_POOL_NAME" \
  --auto-verified-attributes email \
  --username-attributes email \
  --policies '{"PasswordPolicy":
{"MinimumLength":12,"RequireUppercase":true,"RequireLowercase":true,"RequireNumbers":true
 \
  --schema '[{"Name":"email","Required":true,"Mutable":true}]' \
  --mfa-configuration OFF \
  --user-attribute-update-settings '{"AttributesRequireVerificationBeforeUpdate":
["email"]}' \
  --account-recovery-setting
 'RecoveryMechanisms=[{Name=verified_email,Priority=1}]' \
```

```
  --deletion-protection INACTIVE \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "create-user-pool"

# Extract the User Pool ID using jq for safety
USER_POOL_ID=$(echo "$USER_POOL_OUTPUT" | jq -r '.UserPool.Id // empty')
if [ -z "$USER_POOL_ID" ]; then
  echo "ERROR: Failed to extract User Pool ID" >&2
  exit 1
fi

# Validate User Pool ID format
if ! [[ "$USER_POOL_ID" =~ ^[a-z]{2}-[a-z]+-[0-9]+_[a-zA-Z0-9]+$ ]]; then
  echo "ERROR: Invalid User Pool ID format: $USER_POOL_ID" >&2
  exit 1
fi

echo "User Pool created with ID: $USER_POOL_ID"

USER_POOL_ARN=$(echo "$USER_POOL_OUTPUT" | jq -r '.UserPool.Arn // empty')
aws cognito-idp tag-resource \
  --resource-arn "$USER_POOL_ARN" \
  --tags project=doc-smith,tutorial=amazon-cognito-gs

# Wait for user pool to be ready
echo "Waiting for user pool to be ready..."
sleep 5

# Step 2: Create an App Client with enhanced security
echo "Creating app client..."
APP_CLIENT_OUTPUT=$(aws cognito-idp create-user-pool-client \
  --user-pool-id "$USER_POOL_ID" \
  --client-name "$APP_CLIENT_NAME" \
  --no-generate-secret \
  --explicit-auth-flows ALLOW_REFRESH_TOKEN_AUTH ALLOW_USER_PASSWORD_AUTH \
  --callback-urls '["https://localhost:3000/callback"]' \
  --allowed-o-auth-flows 'code' \
  --allowed-o-auth-scopes 'openid' 'email' 'profile' \
  --allowed-o-auth-flows-user-pool-client \
  --prevent-user-existence-errors ENABLED \
  --enable-token-revocation \
  --access-token-validity 1 \
  --id-token-validity 1 \
```

```
  --refresh-token-validity 30 \
  --token-validity-units 'AccessToken=hours,IdToken=hours,RefreshToken=days' \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "create-user-pool-client"

# Extract the Client ID using jq for safety
CLIENT_ID=$(echo "$APP_CLIENT_OUTPUT" | jq -r '.UserPoolClient.ClientId //
 empty')
if [ -z "$CLIENT_ID" ]; then
  echo "ERROR: Failed to extract Client ID" >&2
  cleanup_on_error
  exit 1
fi

# Validate Client ID format
if ! [[ "$CLIENT_ID" =~ ^[a-z0-9]{26}$ ]]; then
  echo "ERROR: Invalid Client ID format: $CLIENT_ID" >&2
  cleanup_on_error
  exit 1
fi

echo "App Client created with ID: $CLIENT_ID"

# Step 3: Set Up a Domain for Your User Pool
echo "Setting up user pool domain..."
DOMAIN_OUTPUT=$(aws cognito-idp create-user-pool-domain \
  --user-pool-id "$USER_POOL_ID" \
  --domain "$DOMAIN_NAME" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "create-user-pool-domain"
echo "Domain created: $DOMAIN_NAME.auth.$AWS_REGION.amazoncognito.com"

# Step 4: View User Pool Details
echo "Retrieving user pool details..."
USER_POOL_DETAILS=$(aws cognito-idp describe-user-pool \
  --user-pool-id "$USER_POOL_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "describe-user-pool"
echo "User Pool details retrieved successfully"

# Step 5: View App Client Details
```

```
echo "Retrieving app client details..."
APP_CLIENT_DETAILS=$(aws cognito-idp describe-user-pool-client \
  --user-pool-id "$USER_POOL_ID" \
  --client-id "$CLIENT_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "describe-user-pool-client"
echo "App Client details retrieved successfully"

# Step 6: Create a User (Admin)
echo "Creating admin user..."
ADMIN_USER_EMAIL="admin@example.com"
TEMP_PASSWORD="$(openssl rand -base64 12 | tr -d '\n')!@#"
if [ -z "$TEMP_PASSWORD" ]; then
  echo "ERROR: Failed to generate temporary password" >&2
  cleanup_on_error
  exit 1
fi

ADMIN_USER_OUTPUT=$(aws cognito-idp admin-create-user \
  --user-pool-id "$USER_POOL_ID" \
  --username "$ADMIN_USER_EMAIL" \
  --user-attributes Name=email,Value="$ADMIN_USER_EMAIL"
 Name=email_verified,Value=true \
  --temporary-password "$TEMP_PASSWORD" \
  --message-action SUPPRESS \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "admin-create-user"
echo "Admin user created: $ADMIN_USER_EMAIL"

# Securely clear temporary password from memory
unset TEMP_PASSWORD

# Step 7: Self-Registration
echo "Demonstrating self-registration..."
USER_EMAIL="user@example.com"
USER_PASSWORD="SecurePassword123!"
SIGNUP_OUTPUT=$(aws cognito-idp sign-up \
  --client-id "$CLIENT_ID" \
  --username "$USER_EMAIL" \
  --password "$USER_PASSWORD" \
  --user-attributes Name=email,Value="$USER_EMAIL" \
  --region "$AWS_REGION" \
```

```
  2>&1)
check_aws_error "sign-up"
echo "User signed up: $USER_EMAIL"
echo "A confirmation code would be sent to the user's email in a real scenario"

# Securely clear password from memory
unset USER_PASSWORD

echo ""
echo "================================================="
echo "IMPORTANT: In a real scenario, the user would receive"
echo "a confirmation code via email. For this demo, we'll"
echo "use admin-confirm-sign-up instead."
echo "================================================="
echo ""

# Step 8: Confirm User Registration (using admin privileges for demo)
echo "Confirming user registration (admin method)..."
CONFIRM_OUTPUT=$(aws cognito-idp admin-confirm-sign-up \
  --user-pool-id "$USER_POOL_ID" \
  --username "$USER_EMAIL" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "admin-confirm-sign-up"
echo "User confirmed: $USER_EMAIL"

# Step 9: Set permanent password for user
echo "Setting permanent password for user..."
SET_PASSWORD="SecureUserPassword123!"
SET_PASS_OUTPUT=$(aws cognito-idp admin-set-user-password \
  --user-pool-id "$USER_POOL_ID" \
  --username "$USER_EMAIL" \
  --password "$SET_PASSWORD" \
  --permanent \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "admin-set-user-password"
echo "Permanent password set for user"

unset SET_PASSWORD

# Step 10: Authenticate a User
echo "Authenticating user..."
AUTH_PASSWORD="SecureUserPassword123!"
```

```
AUTH_OUTPUT=$(aws cognito-idp initiate-auth \
  --client-id "$CLIENT_ID" \
  --auth-flow USER_PASSWORD_AUTH \
  --auth-parameters "USERNAME=$USER_EMAIL,PASSWORD=$AUTH_PASSWORD" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "initiate-auth"
echo "User authenticated successfully"

unset AUTH_PASSWORD

# Extract auth tokens securely
ID_TOKEN=$(echo "$AUTH_OUTPUT" | jq -r '.AuthenticationResult.IdToken // empty')
ACCESS_TOKEN=$(echo "$AUTH_OUTPUT" | jq -r '.AuthenticationResult.AccessToken //
 empty')

# Validate tokens exist
if [ -z "$ID_TOKEN" ] || [ -z "$ACCESS_TOKEN" ]; then
  echo "WARNING: Failed to extract authentication tokens" >&2
else
  echo "Authentication tokens obtained successfully"
fi

# Securely clear tokens from memory
unset ID_TOKEN
unset ACCESS_TOKEN

# Step 11: List Users in the User Pool
echo "Listing users in the user pool..."
USERS_OUTPUT=$(aws cognito-idp list-users \
  --user-pool-id "$USER_POOL_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "list-users"
echo "Users listed successfully"

# Display summary of created resources
echo ""
echo "================================================"
echo "RESOURCE SUMMARY"
echo "================================================"
echo "User Pool ID: $USER_POOL_ID"
echo "User Pool Name: $USER_POOL_NAME"
echo "App Client ID: $CLIENT_ID"
```

```
echo "App Client Name: $APP_CLIENT_NAME"
echo "Domain: $DOMAIN_NAME.auth.$AWS_REGION.amazoncognito.com"
echo "Admin User: $ADMIN_USER_EMAIL"
echo "Regular User: $USER_EMAIL"
echo "================================================="
echo ""

# Auto-confirm cleanup
echo ""
echo "=========================================="
echo "CLEANUP"
echo "=========================================="
echo "Starting cleanup process..."

# Step 12: Clean Up Resources
echo "Deleting user pool domain..."
DELETE_DOMAIN_OUTPUT=$(aws cognito-idp delete-user-pool-domain \
  --user-pool-id "$USER_POOL_ID" \
  --domain "$DOMAIN_NAME" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "delete-user-pool-domain"
echo "Domain deleted successfully"

# Wait for domain deletion to complete
echo "Waiting for domain deletion to complete..."
sleep 5

echo "Deleting user pool (this will also delete the app client)..."
DELETE_POOL_OUTPUT=$(aws cognito-idp delete-user-pool \
  --user-pool-id "$USER_POOL_ID" \
  --region "$AWS_REGION" \
  2>&1)
check_aws_error "delete-user-pool"
echo "User pool deleted successfully"

echo "All resources have been cleaned up"

echo "Script completed at $(date)"

# Remove trap to prevent cleanup on successful exit
trap - EXIT ERR
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de comandos de AWS CLI .
  - AdminConfirmSignUp
  - AdminCreateUser
  - CreateUserPool
  - CreateUserPoolClient
  - CreateUserPoolDomain
  - DeleteUserPool
  - DeleteUserPoolDomain
  - DescribeUserPool
  - DescribeUserPoolClient
  - InitiateAuth
  - ListUsers
  - SignUp

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo empezar 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 AWS SDK

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

SDK para .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 detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para .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 detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para 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 detalles 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 { logger } 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);
    logger.log("Signing up.");
    await signUp({ clientId, username, password, email });
    logger.log(`Signed up. A confirmation email has been sent to: ${email}.`);
    logger.log(
      `Run 'confirm-sign-up ${username} <code>' to confirm your account.`,
    );
  } catch (err) {
    logger.error(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 { logger } 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);
    logger.log("Confirming user.");
    await confirmSignUp({ clientId, username, code });
    logger.log(
      `User confirmed. Run 'admin-initiate-auth ${username} <password>' to sign
 in.`,
    );
  } catch (err) {
    logger.error(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 { logger } 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);

    logger.log("Signing in.");
    const { ChallengeName, Session } = await adminInitiateAuth({
      clientId,
      userPoolId,
      username,
      password,
    });

    if (ChallengeName === "MFA_SETUP") {
      logger.log("MFA setup is required.");
      return handleMfaSetup(Session, username);
    }
```

```
    if (ChallengeName === "SOFTWARE_TOKEN_MFA") {
      handleSoftwareTokenMfa(Session);
      logger.log(`Run 'admin-respond-to-auth-challenge ${username} <totp>'`);
    }
  } catch (err) {
    logger.error(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 { logger } 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);

    logger.log("Successfully authenticated.");
  } catch (err) {
    logger.error(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 { logger } 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);

    logger.log("Verifying TOTP.");
    await verifySoftwareToken(totp);
    logger.log("TOTP Verified. Run 'admin-initiate-auth' again to sign-in.");
  } catch (err) {
    logger.error(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 detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para JavaScript .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información 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.fromEnvironment { 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.fromEnvironment { 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.fromEnvironment { 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.fromEnvironment { 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.fromEnvironment { 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.fromEnvironment { 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.fromEnvironment { 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.fromEnvironment { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.signUp(signUpRequest)
        println("User has been signed up")
    }
}
```

- Para obtener detalles sobre 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)
            # start_sign_in may return an AuthenticationResult with no challenge
```

```
            # (for example, when the user pool does not require MFA). Read the
            # challenge defensively and stop if sign-in already succeeded.
            challenge = response.get("ChallengeName")
            if challenge is None:
                auth_tokens = response.get("AuthenticationResult")
                if auth_tokens is not None:
                    print(f"You're signed in as {user_name}.")
                    print("Here's your access token:")
                    pp(auth_tokens["AccessToken"])
                break
        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"])
            if "NewDeviceMetadata" in auth_tokens:
                print("And your device information:")
                pp(auth_tokens["NewDeviceMetadata"])
            challenge = None
        else:
            raise Exception(f"Got unexpected challenge
 {response['ChallengeName']}")
    print("-" * 88)

    # If sign-in returned neither a challenge nor an AuthenticationResult, there
    # are no tokens to work with; nothing more to do.
    if auth_tokens is None:
        return user_name

    # respond_to_mfa_challenge returns NewDeviceMetadata only when the user pool
    # has device tracking configured. When it's absent, skip the device-
confirmation
    # and tracked-device sign-in steps instead of failing on a missing key.
    device_metadata = auth_tokens.get("NewDeviceMetadata")
    if device_metadata is None:
        print(
            "This user pool doesn't track devices, so there's no device to
 confirm. "
            "To try the tracked-device sign-in flow, enable device tracking on
 the "
            "user pool and run this example again."
        )
        print("-" * 88)
    else:
        device_group_key = device_metadata["DeviceGroupKey"]
        device_key = device_metadata["DeviceKey"]
        device_password =
 base64.standard_b64encode(os.urandom(40)).decode("utf-8")

        print(
            "Let's confirm your MFA device so you don't have to 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)
    return user_name


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()
    client = boto3.client("cognito-idp")
    created_user = None
    try:
        created_user = run_scenario(client, args.user_pool_id, args.client_id)
    except Exception:
        logging.exception("Something went wrong with the demo.")
    finally:
        # Clean up the user this scenario created, on success or failure, so a
        # run never leaves an orphaned user in the pool. Cleanup errors are
        # logged but do not mask the original outcome.
        if created_user is not None:
            try:
                client.admin_delete_user(
                    UserPoolId=args.user_pool_id, Username=created_user
                )
                print(f"Deleted user {created_user}.")
            except Exception:
                logging.exception(
                    "Couldn't delete user %s; delete it manually.", created_user
                )


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

Swift

SDK para Swift

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

El archivo `Package.swift`.

```
// swift-tools-version: 5.9
//
// The swift-tools-version declares the minimum version of Swift required to
// build this package.

import PackageDescription

let package = Package(
    name: "cognito-scenario",
    // Let Xcode know the minimum Apple platforms supported.
    platforms: [
        .macOS(.v13),
        .iOS(.v15)
    ],
    dependencies: [
        // Dependencies declare other packages that this package depends on.
        .package(
            url: "https://github.com/awslabs/aws-sdk-swift",
            from: "1.0.0"),
        .package(
            url: "https://github.com/apple/swift-argument-parser.git",
            branch: "main"
        )
    ],
    targets: [
```

```
        // Targets are the basic building blocks of a package, defining a module
 or a test suite.
        // Targets can depend on other targets in this package and products
        // from dependencies.
        .executableTarget(
            name: "cognito-scenario",
            dependencies: [
                .product(name: "AWSCognitoIdentityProvider", package: "aws-sdk-
swift"),
                .product(name: "ArgumentParser", package: "swift-argument-
parser")
            ],
            path: "Sources")

    ]
)
```

El archivo de código de Swift.

```
// An example demonstrating various features of Amazon Cognito. Before running
// this Swift 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 example performs the following functions:
//
// 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.

import ArgumentParser
import Foundation

import AWSClientRuntime
import AWSCognitoIdentityProvider

struct ExampleCommand: ParsableCommand {
    @Argument(help: "The application clientId.")
    var clientId: String
    @Argument(help: "The user pool ID to use.")
    var poolId: String
    @Option(help: "Name of the Amazon Region to use")
    var region = "us-east-1"

    static var configuration = CommandConfiguration(
        commandName: "cognito-scenario",
        abstract: """
        Demonstrates various features of Amazon Cognito.
        """,
        discussion: """
        """
    )

    /// Prompt for an input string of at least a minimum length.
    ///
    /// - Parameters:
    ///   - prompt: The prompt string to display.
    ///   - minLength: The minimum number of characters to allow in the
    ///     response. Default value is 0.
    ///
    /// - Returns: The entered string.
    func stringRequest(_ prompt: String, minLength: Int = 1) -> String {
        while true {
            print(prompt, terminator: "")
            let str = readLine()

            guard let str else {
```

```
                continue
            }
            if str.count >= minLength {
                return str
            } else {
                print("*** Response must be at least \(minLength) character(s)
long.")
            }
        }
    }

    /// Ask a yes/no question.
    ///
    /// - Parameter prompt: A prompt string to print.
    ///
    /// - Returns: `true` if the user answered "Y", otherwise `false`.
    func yesNoRequest(_ prompt: String) -> Bool {
        while true {
            let answer = stringRequest(prompt).lowercased()
            if answer == "y" || answer == "n" {
                return answer == "y"
            }
        }
    }

    /// Get information about a specific user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The Amazon Cognito Identity Provider client to use.
    ///   - userName: The user to retrieve information about.
    ///   - userPoolId: The user pool to search for the specified user.
    ///
    /// - Returns: `true` if the user's information was successfully
    ///   retrieved. Otherwise returns `false`.
    func adminGetUser(cipClient: CognitoIdentityProviderClient, userName: String,
                      userPoolId: String) async -> Bool {
        do {
            let output = try await cipClient.adminGetUser(
                input: AdminGetUserInput(
                    userPoolId: userPoolId,
                    username: userName
                )
            )
```

```
            guard let userStatus = output.userStatus else {
                print("*** Unable to get the user's status.")
                return false
            }

            print("User status: \(userStatus)")
            return true
        } catch {
            return false
        }
    }

    /// Create a new user in a user pool.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The ID of the app client to create a user for.
    ///   - userName: The username for the new user.
    ///   - password: The new user's password.
    ///   - email: The new user's email address.
    ///
    /// - Returns: `true` if successful; otherwise `false`.
    func signUp(cipClient: CognitoIdentityProviderClient, clientId: String,
userName: String, password: String, email: String) async -> Bool {
        let emailAttr = CognitoIdentityProviderClientTypes.AttributeType(
            name: "email",
            value: email
        )

        let userAttrsList = [emailAttr]

        do {
            _ = try await cipClient.signUp(
                input: SignUpInput(
                    clientId: clientId,
                    password: password,
                    userAttributes: userAttrsList,
                    username: userName
                )

            )

            print("=====> User \(userName) signed up.")
        } catch _ as AWSCognitoIdentityProvider.UsernameExistsException {
```

```
            print("*** The username \(userName) already exists. Please use a
different one.")
            return false
        } catch let error as AWSCognitoIdentityProvider.InvalidPasswordException
{
            print("*** Error: The specified password is invalid. Reason:
\(error.properties.message ?? "<none available>").")
            return false
        } catch _ as AWSCognitoIdentityProvider.ResourceNotFoundException {
            print("*** Error: The specified client ID (\(clientId)) doesn't
exist.")
            return false
        } catch {
            print("*** Unexpected error: \(error)")
            return false
        }

        return true
    }

    /// Requests a new confirmation code be sent to the given user's contact
    /// method.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The application client ID.
    ///   - userName: The user to resend a code for.
    ///
    /// - Returns: `true` if a new code was sent successfully, otherwise
    ///   `false`.
    func resendConfirmationCode(cipClient: CognitoIdentityProviderClient,
clientId: String,
                                userName: String) async -> Bool {
        do {
            let output = try await cipClient.resendConfirmationCode(
                input: ResendConfirmationCodeInput(
                    clientId: clientId,
                    username: userName
                )
            )

            guard let deliveryMedium = output.codeDeliveryDetails?.deliveryMedium
else {
```

```
                print("*** Unable to get the delivery method for the resent
code.")
                return false
            }

            print("=====> A new code has been sent by \(deliveryMedium)")
            return true
        } catch {
            print("*** Unable to resend the confirmation code to user
\(userName).")
            return false
        }
    }

    /// Submit a confirmation code for the specified user. This is the code as
    /// entered by the user after they've received it by email or text
    /// message.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID the user is signing up for.
    ///   - userName: The username of the user whose code is being sent.
    ///   - code: The user's confirmation code.
    ///
    /// - Returns: `true` if the code was successfully confirmed; otherwise
`false`.
    func confirmSignUp(cipClient: CognitoIdentityProviderClient, clientId:
String,
                       userName: String, code: String) async -> Bool {
        do {
            _ = try await cipClient.confirmSignUp(
                input: ConfirmSignUpInput(
                    clientId: clientId,
                    confirmationCode: code,
                    username: userName
                )
            )

            print("=====> \(userName) has been confirmed.")
            return true
        } catch {
            print("=====> \(userName)'s code was entered incorrectly.")
            return false
        }
```

```
    }

    /// Begin an authentication session.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongitoIdentityProviderClient` to use.
    ///   - clientId: The app client ID to use.
    ///   - userName: The username to check.
    ///   - password: The user's password.
    ///   - userPoolId: The user pool to use.
    ///
    /// - Returns: The session token associated with this authentication
    ///   session.
    func initiateAuth(cipClient: CognitoIdentityProviderClient, clientId: String,
                      userName: String, password: String,
                      userPoolId: String) async -> String? {
        var authParams: [String: String] = [:]

        authParams["USERNAME"] = userName
        authParams["PASSWORD"] = password

        do {
            let output = try await cipClient.adminInitiateAuth(
                input: AdminInitiateAuthInput(
                    authFlow:
CognitoIdentityProviderClientTypes.AuthFlowType.adminUserPasswordAuth,
                    authParameters: authParams,
                    clientId: clientId,
                    userPoolId: userPoolId
                )
            )

            guard let challengeName = output.challengeName else {
                print("*** Invalid response from the auth service.")
                return nil
            }

            print("=====> Response challenge is \(challengeName)")

            return output.session
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return nil
        } catch _ as UserNotConfirmedException {
```

```
            print("*** The user \(userName) has not been confirmed.")
            return nil
        } catch {
            print("*** An unexpected error occurred.")
            return nil
        }
    }

    /// Request and display an MFA secret token that the user should enter
    /// into their authenticator to set it up for the user account.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - authSession: The authentication session to request an MFA secret
    ///     for.
    ///
    /// - Returns: A string containing the MFA secret token that should be
    ///   entered into the authenticator software.
    func getSecretForAppMFA(cipClient: CognitoIdentityProviderClient,
authSession: String?) async -> String? {
        do {
            let output = try await cipClient.associateSoftwareToken(
                input: AssociateSoftwareTokenInput(
                    session: authSession
                )
            )

            guard let secretCode = output.secretCode else {
                print("*** Unable to get the secret code")
                return nil
            }

            print("=====> Enter this token into Google Authenticator:
\(secretCode)")
            return output.session
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return nil
        } catch {
            print("*** An unexpected error occurred getting the secret for the
app's MFA.")
            return nil
        }
    }
```

```
    /// Confirm that the user's TOTP authenticator is configured correctly by
    /// sending a code to it to check that it matches successfully.
    ///
    /// - Parameters:
    ///   - cipClient: The `CongnitoIdentityProviderClient` to use.
    ///   - session: An authentication session previously returned by an
    ///     `associateSoftwareToken()` call.
    ///   - mfaCode: The 6-digit code currently displayed by the user's
    ///     authenticator, as provided by the user.
    func verifyTOTP(cipClient: CognitoIdentityProviderClient, session: String?,
mfaCode: String?) async {
        do {
            let output = try await cipClient.verifySoftwareToken(
                input: VerifySoftwareTokenInput(
                    session: session,
                    userCode: mfaCode
                )
            )

            guard let tokenStatus = output.status else {
                print("*** Unable to get the token's status.")
                return
            }
            print("=====> The token's status is: \(tokenStatus)")
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user has not been confirmed.")
            return
        } catch {
            print("*** Error verifying the MFA token!")
            return
        }
    }

    /// Respond to the authentication challenge received from Cognito after
```

```
    /// initiating an authentication session. This involves sending a current
    /// MFA code to the service.
    ///
    /// - Parameters:
    ///   - cipClient: The `CognitoIdentityProviderClient` to use.
    ///   - userName: The user's username.
    ///   - clientId: The app client ID.
    ///   - userPoolId: The user pool to sign into.
    ///   - mfaCode: The 6-digit MFA code currently displayed by the user's
    ///     authenticator.
    ///   - session: The authentication session to continue processing.
    func adminRespondToAuthChallenge(cipClient: CognitoIdentityProviderClient,
userName: String,
                                     clientId: String, userPoolId: String,
mfaCode: String,
                                     session: String) async {
        print("=====> SOFTWARE_TOKEN_MFA challenge is generated...")

        var challengeResponsesOb: [String: String] = [:]
        challengeResponsesOb["USERNAME"] = userName
        challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

        do {
            let output = try await cipClient.adminRespondToAuthChallenge(
                input: AdminRespondToAuthChallengeInput(
                    challengeName:
CognitoIdentityProviderClientTypes.ChallengeNameType.softwareTokenMfa,
                    challengeResponses: challengeResponsesOb,
                    clientId: clientId,
                    session: session,
                    userPoolId: userPoolId
                )
            )

            guard let authenticationResult = output.authenticationResult else {
                print("*** Unable to get authentication result.")
                return
            }

            print("=====> Authentication result (JWTs are redacted):")
            print(authenticationResult)
        } catch _ as SoftwareTokenMFANotFoundException {
            print("*** The specified user pool isn't configured for MFA.")
            return
```

```
        } catch _ as CodeMismatchException {
            print("*** The specified MFA code doesn't match the expected value.")
            return
        } catch _ as UserNotFoundException {
            print("*** The specified username, \(userName), doesn't exist.")
            return
        } catch _ as UserNotConfirmedException {
            print("*** The user \(userName) has not been confirmed.")
            return
        } catch let error as NotAuthorizedException {
            print("*** Unauthorized access. Reason: \(error.properties.message ??
"<unknown>")")
        } catch {
            print("*** Error responding to the MFA challenge.")
            return
        }
    }

    /// Called by ``main()`` to run the bulk of the example.
    func runAsync() async throws {
        let config = try await
CognitoIdentityProviderClient.CognitoIdentityProviderClientConfig(region:
region)
        let cipClient = CognitoIdentityProviderClient(config: config)

        print("""
              This example collects information about a user, then creates that
user in the
              specified user pool. Then, it enables Multi-Factor Authentication
(MFA) for that
              user by associating an authenticator application (such as Google
Authenticator
              or a password manager that supports TOTP). Then, the user uses a
code from their
              authenticator application to sign in.

              """)

        let userName = stringRequest("Please enter a new username: ")
        let password = stringRequest("Enter a password: ")
        let email = stringRequest("Enter your email address: ", minLength: 5)

        // Submit the sign-up request to AWS.
```

```
        print("==> Signing up user \(userName)...")
        if await signUp(cipClient: cipClient, clientId: clientId,
                        userName: userName, password: password,
                        email: email) == false {
            return
        }

        // Check the user's status. This time, it should come back "unconfirmed".

        print("==> Getting the status of user \(userName) from the user pool
(should be 'unconfirmed')...")
        if await adminGetUser(cipClient: cipClient, userName: userName,
userPoolId: poolId) == false {
            return
        }

        // Ask the user if they want a replacement code sent, such as if the
        // code hasn't arrived yet. If the user responds with a "yes," send a
        // new code.

        if yesNoRequest("==> A confirmation code was sent to \(userName). Would
you like to send a new code (Y/N)? ") {
            print("==> Sending a new confirmation code...")
            if await resendConfirmationCode(cipClient: cipClient, clientId:
clientId, userName: userName) == false {
                return
            }
        }

        // Ask the user to enter the confirmation code, then send it to Amazon
        // Cognito to verify it.

        let code = stringRequest("==> Enter the confirmation code sent to
\(userName): ")
        if await confirmSignUp(cipClient: cipClient, clientId: clientId,
userName: userName, code: code) == false {
            // The code didn't match. Your application may wish to offer to
            // re-send the confirmation code here and try again.
            return
        }

        // Check the user's status again. This time it should come back
        // "confirmed".
```

```
        print("==> Rechecking status of user \(userName) in the user pool (should
be 'confirmed')...")
        if await adminGetUser(cipClient: cipClient, userName: userName,
userPoolId: poolId) == false {
            return
        }
        // Check the challenge mode. Here, it should be "mfaSetup", indicating
        // that the user needs to add MFA before using it. This returns a
        // session that can be used to register MFA, or nil if an error occurs.

        let authSession = await initiateAuth(cipClient: cipClient, clientId:
clientId,
                                             userName: userName, password:
password,
                                             userPoolId: poolId)
        if authSession == nil {
            return
        }

        // Ask Cognito for an MFA secret token that the user should enter into
        // their authenticator software (such as Google Authenticator) or
        // password manager to configure it for this user account. This
        // returns a new session that should be used for the new stage of the
        // authentication process.

        let newSession = await getSecretForAppMFA(cipClient: cipClient,
authSession: authSession)
        if newSession == nil {
            return
        }

        // Ask the user to enter the current 6-digit code displayed by their
        // authenticator. Then verify that it matches the value expected for
        // the session.

        let mfaCode1 = stringRequest("==> Enter the 6-digit code displayed in
your authenticator: ",
                                     minLength: 6)
        await verifyTOTP(cipClient: cipClient, session: newSession, mfaCode:
mfaCode1)

        // Ask the user to authenticate now that the authenticator has been
        // configured. This creates a new session using the user's username
        // and password as already entered.
```

```
        print("\nNow starting the sign-in process for user \(userName)...\n")

        let session2 = await initiateAuth(cipClient: cipClient, clientId:
 clientId,
                                          userName: userName, password: password,
 userPoolId: poolId)
        guard let session2 else {
            return
        }

        // Now that we have a new auth session, `session2`, ask the user for a
        // new 6-digit code from their authenticator, and send it to the auth
        // session.

        let mfaCode2 = stringRequest("==> Wait for your authenticator to show a
 new 6-digit code, then enter it: ",
                                     minLength: 6)
        await adminRespondToAuthChallenge(cipClient: cipClient, userName:
 userName,
                                          clientId: clientId, userPoolId: poolId,
                                          mfaCode: mfaCode2, session: session2)
    }
}

/// The program's asynchronous entry point.
@main
struct Main {
    static func main() async {
        let args = Array(CommandLine.arguments.dropFirst())

        do {
            let command = try ExampleCommand.parse(args)
            try await command.runAsync()
        } catch {
            ExampleCommand.exit(withError: error)
        }
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK para Swift.

- 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 AWS SDK. En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de los grupos de identidades y los flujos de identidades de Amazon Cognito

El siguiente ejemplo de código muestra cómo crear una aplicación de demostración basada en web que muestre los flujos de autenticación de los grupos de identidades.

Python

SDK para Python (Boto3)

Podemos ver una aplicación de demostración basada en web que muestra los flujos de autenticación de los grupos de identidades de Amazon Cognito, lo que permite a los usuarios explorar de forma interactiva los flujos de autenticación básicos y mejorados con varios proveedores de identidad.

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

Servicios utilizados en este ejemplo

- Amazon Cognito Identity Provider

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo empezar 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 AWS SDK

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.

```
import (
 "context"
 "errors"
 "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"
)

// 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(ctx context.Context, 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(ctx, 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(ctx, 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(ctx, 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(ctx context.Context, 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(
  ctx, 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(ctx context.Context, 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(ctx, 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(ctx context.Context, 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(ctx, 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(ctx context.Context, stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup(ctx)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(ctx, stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]
 runner.helper.PopulateUserTable(ctx, stackOutputs["TableName"])
 userName, password := runner.AddUserToPool(ctx, stackOutputs["UserPoolId"],
 stackOutputs["TableName"])

 runner.AddActivityLogTrigger(ctx, stackOutputs["UserPoolId"],
 stackOutputs["ActivityLogFunctionArn"])
 runner.SignInUser(ctx, stackOutputs["UserPoolClientId"], userName, password)
 runner.helper.ListRecentLogEvents(ctx, stackOutputs["ActivityLogFunction"])
 runner.GetKnownUserLastLogin(ctx, stackOutputs["TableName"], userName)

 runner.resources.Cleanup(ctx)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Controle el desencadenador `PostAuthentication` con una función de Lambda.

```
import (
 "context"
 "fmt"
 "log"
 "os"
 "time"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 dynamodbtypes "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

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() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 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.

```
import (
 "context"
 "log"
 "strings"
 "time"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cloudformation"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
 GetStackOutputs(ctx context.Context, stackName string) (actions.StackOutputs,
 error)
 PopulateUserTable(ctx context.Context, tableName string)
 GetKnownUsers(ctx context.Context, tableName string) (actions.UserList, error)
 AddKnownUser(ctx context.Context, tableName string, user actions.User)
```

```
 ListRecentLogEvents(ctx context.Context, 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(ctx context.Context, stackName
 string) (actions.StackOutputs, error) {
 return helper.cfnActor.GetOutputs(ctx, stackName), nil
}

// PopulateUserTable fills the known user table with example data.
```

```
func (helper ScenarioHelper) PopulateUserTable(ctx context.Context, 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(ctx, tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(ctx context.Context, tableName string)
 (actions.UserList, error) {
 knownUsers, err := helper.dynamoActor.Scan(ctx, 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(ctx context.Context, 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(ctx, 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(ctx context.Context,
 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(ctx, functionName)
 if err != nil {
```

```
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(ctx, 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.

```
import (
 "context"
 "errors"
 "log"

 "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"
)

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(ctx context.Context, userPoolId
 string, triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(ctx,
 &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(ctx,
 &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(ctx context.Context, clientId string, userName
 string, password string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(ctx,
 &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(ctx context.Context, clientId string, userName
 string, password string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(ctx,
 &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(ctx context.Context, clientId string,
 userName string) (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(ctx,
 &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(ctx context.Context, clientId
 string, code string, userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(ctx,
 &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(ctx context.Context, userAccessToken
 string) error {
 _, err := actor.CognitoClient.DeleteUser(ctx,
 &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(ctx context.Context, userPoolId
 string, userName string, userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(ctx,
 &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(ctx context.Context, userPoolId
 string, userName string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(ctx,
 &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.

```
import (
 "context"
 "fmt"
 "log"
```

```
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/dynamodb/attributevalue"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb"
 "github.com/aws/aws-sdk-go-v2/service/dynamodb/types"
)

// 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(ctx context.Context, 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(ctx, &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(ctx context.Context, tableName string) (UserList,
 error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(ctx, &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(ctx context.Context, 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(ctx, &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 abarque las acciones de CloudWatch Logs.

```
import (
 "context"
 "fmt"
 "log"

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

type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(ctx context.Context,
 functionName string) (types.LogStream, error) {
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
```

```
 output, err := actor.CwlClient.DescribeLogStreams(ctx,
 &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(ctx context.Context, 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(ctx,
 &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. CloudFormation

```
import (
 "context"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/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(ctx context.Context, stackName
 string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(ctx,
 &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.

```
import (
 "context"
 "log"
 "user_pools_and_lambda_triggers/actions"
```

```
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// 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(ctx context.Context) {
 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(ctx, 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(ctx, 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 detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para 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 AWS SDK En este tema también se incluye información sobre cómo empezar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código para Amazon Cognito Sync mediante AWS SDKs

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 distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

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

Amazon Cognito Sync

- Ejemplos básicos de Amazon Cognito Sync mediante AWS SDKs
  - Acciones para Amazon Cognito Sync mediante AWS SDKs
    - ListIdentityPoolUsageÚselo con un AWS SDK

Ejemplos básicos de Amazon Cognito Sync mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los aspectos básicos de Amazon Cognito Sync con. AWS SDKs

Ejemplos

- Acciones para Amazon Cognito Sync mediante AWS SDKs
  - ListIdentityPoolUsageÚselo con un AWS SDK

Acciones para Amazon Cognito Sync mediante AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Cognito Sync con. AWS SDKs 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

- ListIdentityPoolUsageÚselo con un AWS SDK

`ListIdentityPoolUsage`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `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 AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Otras capacidades

Además de iniciar sesión a los usuarios en su aplicación con un grupo de usuarios, Amazon Cognito ofrece funciones adicionales para necesidades especializadas. Los grupos de identidades de Amazon Cognito intercambian la identidad de un usuario por AWS credenciales temporales con privilegios limitados para que la aplicación pueda llamar a los servicios directamente. AWS

Temas

- Grupos de identidades de Amazon Cognito
- Introducción a los grupos de identidades de Amazon Cognito

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 proveedores de identidad de OIDC para sus aplicaciones. SSO-enabled 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 IdP de grupos de identidades, Configuración de Facebook como un IdP de grupos de identidades, Configuración de

Google como IdP de grupo de identidades, Configuración de Inicio de sesión con Apple como IdP de grupo de identidades, Twitter.

- Grupos de usuarios de Amazon Cognito
- Configuración de un proveedor OIDC como IdP de grupo de identidades
- Configuración de un proveedor SAML como IdP de grupo de identidades
- Developer-authenticated 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

- Información general de la consola de grupos de identidades
- Flujo de autenticación de grupos de identidades
- Roles de IAM
- Prácticas recomendadas de seguridad para los grupos de identidades de Amazon Cognito
- Uso de atributos para el control de acceso
- Uso del control de acceso basado en roles
- Obtención de credenciales
- Acceso Servicios de AWS con credenciales temporales
- Proveedores de identidades de terceros de grupos de identidades
- Developer-authenticated identidades
- Transición de usuarios sin autenticar a usuarios autenticados

Información general de la consola de grupos de identidades

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 identificadores de usuario vinculados a los proveedores de identidades externos.

Para comprender las características y las opciones de los grupos de identidades, una buena opción consiste en crear un grupo de identidades en la consola de Amazon Cognito. Puede observar el efecto de las distintas configuraciones en los flujos de autenticación, el control de acceso basado en roles y atributos y el acceso de invitados. A partir de ahí, puede pasar a estudiar los capítulos posteriores de esta guía y añadir los componentes adecuados para su aplicación para poder implementar la autenticación del grupo de identidades.

Temas

- Creación de un grupo de identidades .
- Roles de IAM de usuario
- Identidades autenticadas y sin autenticar
- Activar o desactivar el acceso de invitados
- Cambio del rol asociado a un tipo de identidad
- Editar proveedores de identidad
- Eliminación de un grupo de identidades
- Eliminación de una identidad de un grupo de identidades

Creación de un grupo de identidades .

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 asigna a su nueva función 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 función con reglas, introduzca la afirmación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que coincida 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. 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, puedes omitir los roles que seleccionaste para ti IdPs y llamar AssumeRoleWithWebIdentitydirectamente. Para obtener más información, consulte Flujo de autenticación de grupos de identidades.
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 asumen los roles que usted crea. 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 identidades de terceros de grupos de identidades.
- Para configurar su propio proceso de autenticación de fondo, consulte. Developer-authenticated 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 su aplicación permite que los usuarios no inicien sesión, puede activar el acceso para las identidades no autenticadas. 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 asigna a su nueva función 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 los usuarios autenticados por 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 desarrollador (su propio proceso de autenticación de back-end). 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 puede acceder ese rol. Servicios de AWS 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 asigna a su nueva función 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 identidades de terceros 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 ha elegido Elegir función con reglas, introduzca la reclamación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que coincida 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. 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. Al eliminar un grupo de identidades, las aplicaciones y otros servicios que utilizan el grupo de identidades dejan 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 desee eliminar y elija Eliminar. Confirme que desea eliminar las identidades y elija Eliminar.

Flujo de autenticación de grupos de identidades

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
Grupo de usuarios de Amazon Cognito	Token de ID
OpenID Connect (OIDC)	Token de ID
SAML 2.0	Aserción de SAML
Proveedor de redes sociales	Token de acceso

El flujo de autenticación mejorado (simplificado)

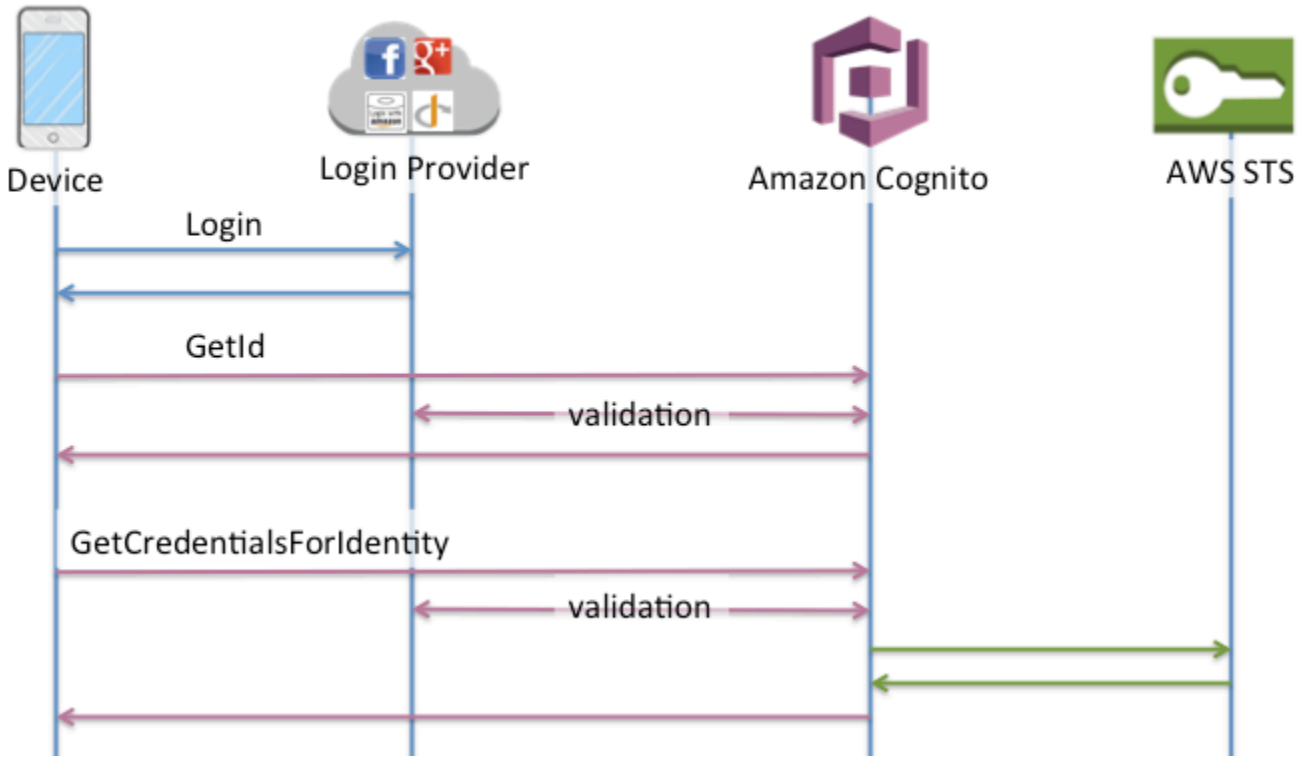
Cuando utilizas el flujo de autenticación mejorado, tu aplicación presenta primero en una solicitud una prueba de autenticación de un grupo de usuarios autorizado de Amazon Cognito o de un proveedor de identidad externo. GetId

1. La aplicación presenta una prueba de autenticación (un token web JSON o una aserción de SAML) de un grupo de usuarios de Amazon Cognito autorizado o de un proveedor de identidades de terceros en una solicitud GetID.
2. El grupo de identidades devuelve un ID 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 administra la lógica de selección de los roles de IAM y la recuperación de credenciales en la configuración del grupo de identidades. Puede configurar el grupo de identidades para que seleccione un rol predeterminado y para que aplique a la selección de roles los principios de control de acceso basado en atributos (ABAC) o de control de acceso basado en roles (RBAC). Las AWS credenciales de la autenticación mejorada son válidas durante una hora.

Orden de las operaciones en la autenticación mejorada

1. `GetId`
2. `GetCredentialsForIdentity`



El flujo de autenticación básico (clásico)

Cuando implementa el flujo de autenticación básico, la aplicación selecciona el rol de IAM que desea que asuman los usuarios.

1. La aplicación presenta una prueba de autenticación (un token web JSON o una aserción de SAML) de un grupo de usuarios de Amazon Cognito autorizado o de un proveedor de identidades de terceros en una solicitud GetID.
2. El grupo de identidades devuelve un ID 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 el grupo de identidades.
5. La 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.

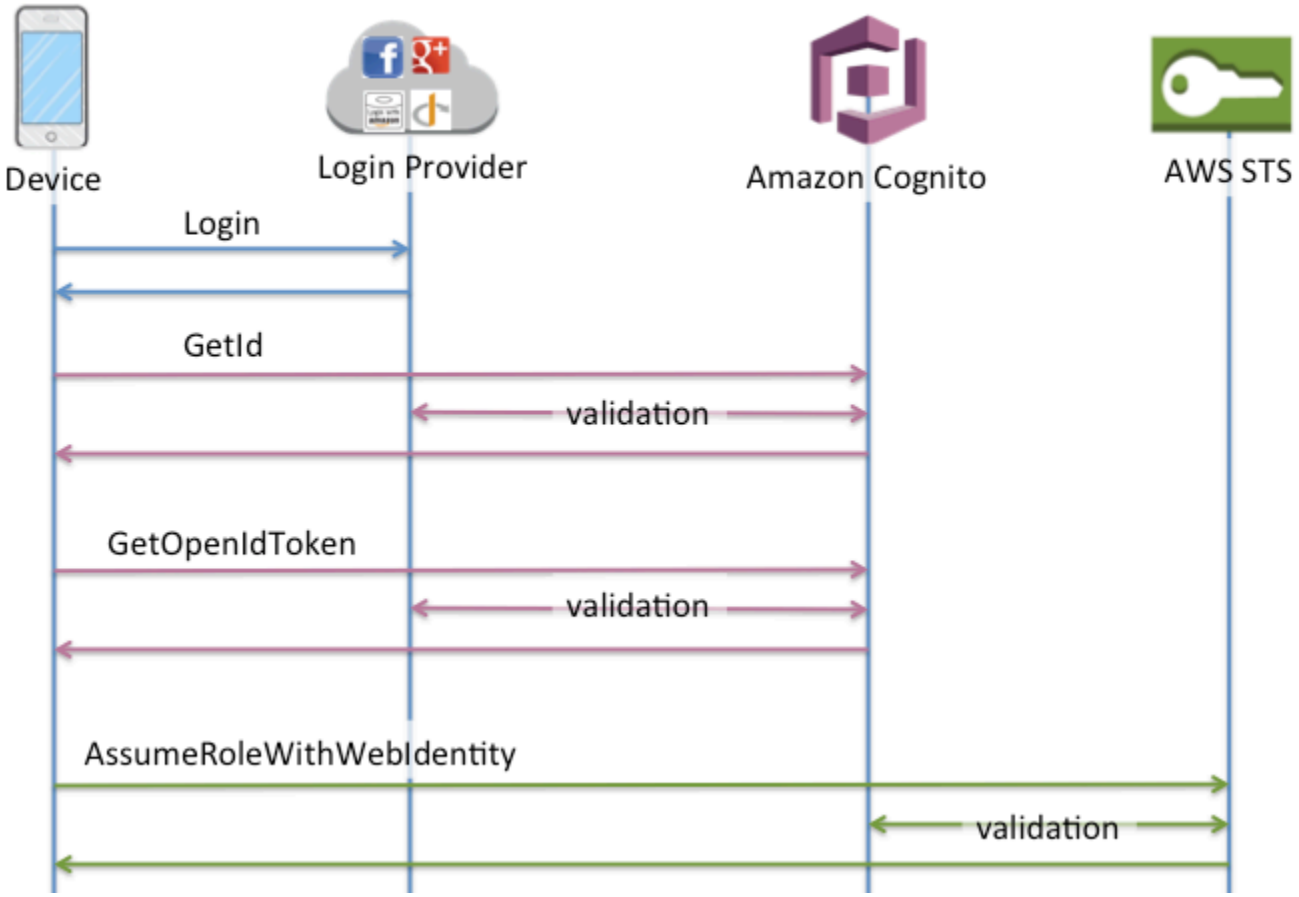
El flujo de trabajo básico le proporciona un control más pormenorizado sobre las credenciales que distribuye a los usuarios. La solicitud `GetCredentialsForIdentity` del flujo de autenticación mejorado solicita un rol basado en el contenido de un token de acceso. La `AssumeRoleWithWebIdentity` solicitud 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.

Puede iniciar sesión con el flujo de autenticación básico en los grupos de usuarios que no tengan asignaciones de roles. Este tipo de grupo de identidades no tiene un rol autenticado o no autenticado predeterminado, y no tiene configurado el control de acceso basado en roles o atributos. Si intenta utilizar `GetOpenIdToken` en un grupo de identidades con asignaciones de roles, recibirá el siguiente error.

```
Basic (classic) flow is not supported with RoleMappings, please use enhanced flow.
```

Orden de las operaciones en la autenticación básica

1. `GetId`
2. `GetOpenIdToken`
3. `AssumeRoleWithWebIdentity`

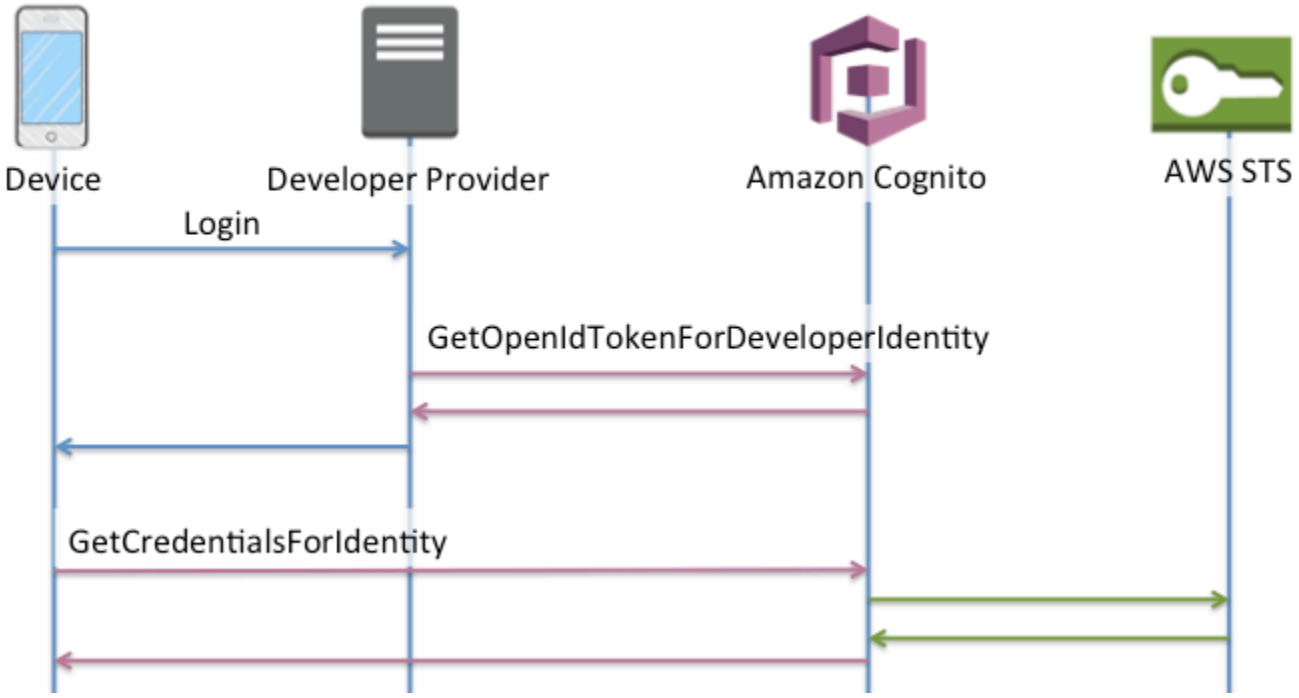


El flujo de autenticación autenticado por el desarrollador

Al utilizar Developer-authenticated identidades, su cliente emplea otro flujo de autenticación, que incluye código ajeno a Amazon Cognito para validar al usuario en su propio sistema de autenticación. Desde la perspectiva de su grupo de identidades, las reclamaciones que presenta en su solicitud de identidad son identificadores arbitrarios, y la autenticación se autoriza mediante las credenciales de IAM que usted codifica en su aplicación.

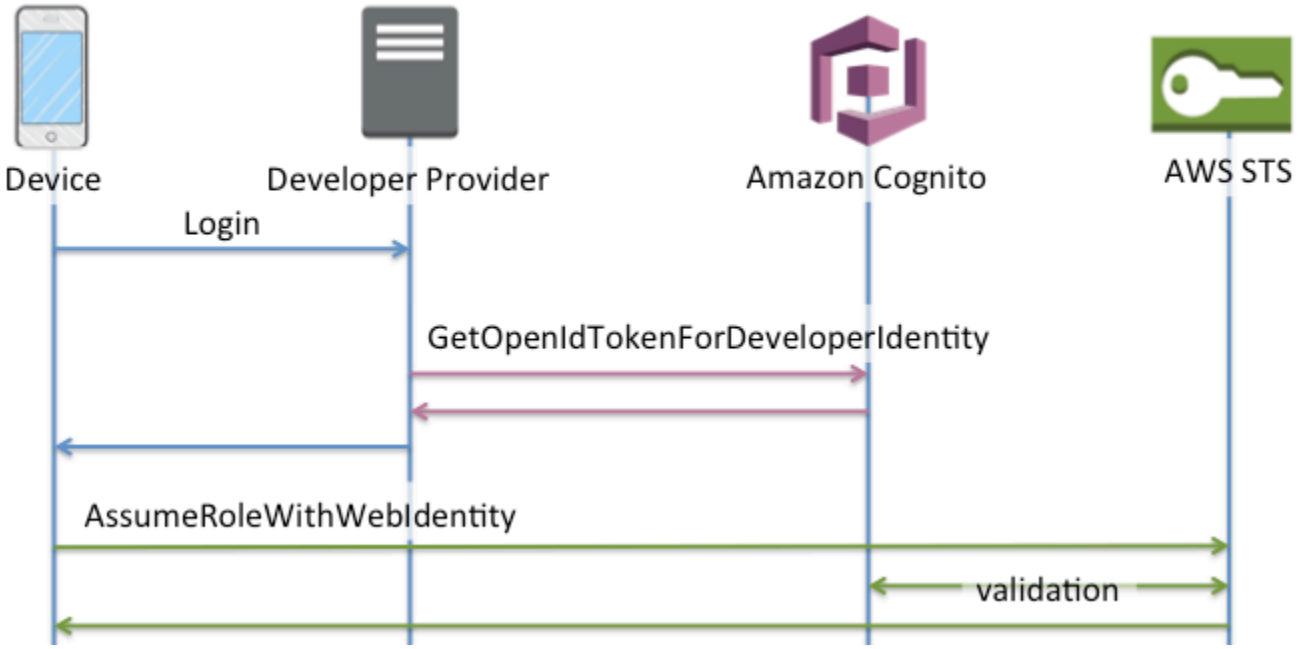
Orden de las operaciones en la autenticación mejorada con un proveedor de desarrollador

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 de 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 en el 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 implementar?

El flujo mejorado es la opción más segura y que exige menos esfuerzo al desarrollador:

- El flujo mejorado reduce la complejidad, el volumen y la velocidad de las solicitudes de API.
- No es necesario que la aplicación realice solicitudes de API adicionales a AWS STS.
- El grupo de identidades evalúa a los usuarios para determinar las credenciales de rol de IAM que deben recibir. No es necesario incorporar en el cliente la lógica de selección de roles.

> **Important**
>
> Como práctica recomendada, cuando cree un grupo de identidades nuevo, no active la autenticación básica (clásica) de forma predeterminada. Para implementar una autenticación básica, primero debe evaluar las relaciones de confianza de los roles de IAM para las identidades web. A continuación, incorpore la lógica de selección de roles al cliente y proteja a este contra cualquier modificación que intenten los usuarios.

El flujo de autenticación básico delega la lógica de selección del rol de IAM a la 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 grupo de identidades o estado. `amr` authenticated/unauthenticated

Tenga también en cuenta que la autenticación del desarrollador es un método abreviado para validar la autenticación del proveedor de identidades. 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 del desarrollador para que los usuarios no puedan acceder a ellos.

Resumen de las operaciones de la API del flujo de autenticación

GetId

La llamada a la API GetId 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 hacer una llamada a la API GetCredentialsForIdentity después de establecer un ID de identidad. Por lo tanto, esta operación es funcionalmente equivalente a llamar GetOpenIdToken. AssumeRoleWithWebIdentity

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 de Amazon Cognito o manualmente mediante la operación SetIdentityPoolRoles.

GetOpenIdToken

Realice una solicitud a la API GetOpenIdToken después de establecer un ID 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. Puede enviar este token como parámetro `WebIdentityToken` en una solicitud AssumeRoleWithWebIdentity.

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 fusionada, una identidad pasa a ser la parent/owner de todos los inicios de sesión asociados y la otra queda deshabilitada. En este caso, se devuelve el parent/owner identificador de. 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 usted.

GetOpenIdTokenForDeveloperIdentity

La GetOpenIdTokenForDeveloperIdentityoperación reemplaza el uso GetOpenIdTokendesde GetIdy desde el dispositivo cuando se utilizan identidades autenticadas por el desarrollador. Dado que la aplicación firma las solicitudes a esta operación de API con credenciales de AWS , Amazon Cognito confía en que el identificador de usuario suministrado en la solicitud sea válido. La autenticación de desarrollador sustituye la validación de token que Amazon Cognito realiza con los proveedores externos.

La carga útil de esta API incluye una asignación `logins`. Esta asignación debe contener la clave del proveedor de desarrollador y un valor para el identificador del usuario en su 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

En cuanto a las identidades que autentica el desarrollador, Amazon Cognito es compatible con la combinación implícita y explícita mediante 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. Debe configurar roles de

IAM diferentes para la aplicación: uno para los usuarios sin autenticar y otro para los usuarios autenticados. La consola de Amazon Cognito puede crear roles predeterminados la primera vez que configure el grupo de identidades. Estos roles no tienen en la práctica permisos concedidos. Modifíquelos para adaptarlos 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 (US-West)	`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`
Opt-in Regiones 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 final de OIDC-standard detección `<issuer>/.well-known/openid-configuration` muestra una ruta al jwks_uri de tu 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. Los roles de IAM funcionan de la siguiente manera: cuando un usuario inicia sesión en su aplicación, Amazon Cognito le genera credenciales temporales de AWS . Estas credenciales temporales se asocian a un rol de IAM específico. Con el rol de IAM, se puede definir un conjunto de permisos para acceder a los recursos de 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 los usuarios que no inician sesión en su aplicación asumen identidades no autenticadas. 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
- Confianza y permisos de rol

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. Amazon Cognito gestiona las conexiones entre un grupo AWS STS de identidades. IdPs

El token que se presenta lo genera un grupo de identidades, que traduce un grupo de usuarios, una red social o un token de 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.

Si la `Principal` de la política de confianza de un rol de IAM es una entidad principal de servicio de grupos de identidades, como `cognito-identity.amazonaws.com`, no será posible crear ni modificar políticas de confianza de roles de tal modo que cualquier grupo de identidades pueda asumir el rol. Con la entidad principal del grupo de identidades, el elemento `Action` debe tener una `Condition` que requiera que solo sus grupos de identidades puedan ejecutar `AssumeRoleWithWebIndentity`, tal y como esté especificado por una clave de condición como `cognito-identity.amazonaws.com:aud`. Hay otras claves de condición disponibles, pero `aud` es obligatoria. Si intenta guardar una política de confianza de roles sin una condición de este tipo, IAM devuelve un error.

Para obtener más información sobre las claves de federación de OIDC (identidad web), consulte Claves disponibles para la federación de OIDC. AWS

A continuación, puede ver claves de condición de federación de OIDC disponibles para Amazon Cognito.

`cognito-identity.amazonaws.com:aud`

Restringe el rol a las operaciones de uno o más grupos de identidades. Amazon Cognito indica el conjunto de identidades de origen en la notificación `aud` del token del grupo de identidades.

`cognito-identity.amazonaws.com:amr`

Restringe el rol a usuarios `authenticated` o `unauthenticated` (invitados). Amazon Cognito indica el estado de la autenticación en la notificación `amr` del token del grupo de identidades.

`cognito-identity.amazonaws.com:sub`

Restringe el rol a uno o más usuarios mediante el UUID. Este UUID es el ID de identidad del usuario del grupo de identidades. No se trata del valor `sub` del proveedor de identidades original del usuario. Amazon Cognito indica este UUID en la notificación `sub` del token 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.

JSON

```
{
    "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 roles de IAM en la autenticación básica (clásica)

Resumen

Los grupos de identidades solo pueden asumir roles en nombre de los usuarios en el flujo de autenticación básica cuando la política de confianza del rol de destino contenga la condición aud.

La autenticación básica tiene la misma limitación sobre políticas de confianza de roles no seguras que la autenticación mejorada: no se puede guardar una política de confianza de roles que no limite los grupos de identidades compatibles con una condición aud. Esta limitación no se aplicaba cuando se lanzó el servicio. Antes de aplicar este requisito, se podían crear políticas de confianza de roles que no tuvieran condiciones de seguridad adicionales. Tras la aplicación de este requisito, AWS STS permite que las identidades web asuman funciones que no estén protegidas con condiciones, pero esas funciones no se pueden modificar sin introducir dichas condiciones.

Enhanced-flow la autenticación requiere que la función de IAM sea la Cuenta de AWS misma que la del grupo de identidades. Sin embargo, en la autenticación básica, en la que la aplicación crea la solicitud `AssumeRoleWithWebIdentity`, la aplicación puede solicitar que se asuma un rol en otra cuenta. Sin embargo, su solicitud de adopción de rol entre cuentas no será válida si el rol de destino tiene una política de confianza heredada que no aplique la condición aud.

El token que un grupo de identidades emite para una identidad contiene información sobre el origen Cuenta de AWS del grupo 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 la adopción de rol fallará 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 los roles del grupo de identidades.

A continuación, se muestra un ejemplo de política de confianza que cumple los requisitos mínimos de un rol de IAM para la autenticación básica con varios grupos de identidades. Además, una práctica recomendada consiste en otorgar permiso solo a las identidades autenticadas con una condición `"cognito-identity.amazonaws.com:amr": "authenticated"`.

JSON

```
{
    "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",
                        "us-west-2:hijklmo-5678-9101-112b-0e4221776g96",
                        "us-west-2:pqrstuv-9101-1121-314c-0e2110887h97"
                    ]
                }
            }
        }
    ]
}
```

Condiciones para la política de confianza adicionales

Puede utilizar las siguientes condiciones de la política de confianza para definir los grupos de identidades, las identidades y los proveedores de origen que pueden asumir roles de IAM.

> **Note**
>
> No implementes la aws:SourceIpcondición clave en las políticas de confianza para las funciones de IAM que asumen los grupos de identidades en el flujo de autenticación mejorada. Como el flujo mejorado genera la AssumeRoleWithWebIdentitysolicitud en nombre de la aplicación, la IP de origen de la solicitud no será la IP del cliente de la aplicación y nunca se cumplirá la condición. Network-based las claves de condición son válidas para las funciones que asumen los grupos de identidades únicamente en el flujo básico, que carece de las funciones de servicio del flujo mejorado.

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 hayan iniciado sesión con un proveedor específico (quizás su propio proveedor de inicio 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]
Aplicaciones empresariales	Amazon Mobile Analytics Amazon Simple Email Service
Computación	AWS Lambda
Criptografía e PKI	AWS Key Management Service[1]
Base de datos	Amazon DynamoDB Amazon SimpleDB
Front-end Web y móvil	AWS AppSync Amazon Location Service Amazon Simple Notification Service Amazon Pinpoint Amazon Location Service
Desarrollo de juegos	GameLift Servidores Amazon
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 AI[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`
	`ReEncryptTo`
	`ReEncryptFrom`
	`GenerateDataKey`
	`GenerateDataKeyPair`
	`GenerateDataKeyPair`
	`GenerateDataKeyPairWithoutPlaintext`
	`GenerateDataKeyWithoutPlaintext`

Amazon SageMaker AI	`InvokeEndpoint`
Amazon Textract	`DetectDocumentText` `AnalyzeDocument`
Amazon Sumerian	`View*`

Amazon Location Service	`SearchPlaceIndex*`
	`GetPlace`
	`CalculateRoute*`
	`*Geofence`
	`*Geofences`
	`*DevicePosition*`

Para conceder acceso Servicios de AWS más allá de esta lista, active el flujo de autenticación básico (clásico) en su grupo de identidades. Si sus usuarios ven `NotAuthorizedException` errores permitidos por las políticas asignadas a la función de IAM para los usuarios no autenticados, evalúe si puede eliminar ese servicio de su caso de uso. Servicios de AWS Si no puede, cambie al flujo básico.

La política de sesión en línea para usuarios invitados

Amazon Cognito aplica primero una política en línea en la solicitud de credenciales de IAM. La política de sesión en línea restringe los permisos efectivos de su usuario para que no tenga acceso a los Servicios de AWS que no aparezcan en 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. Para obtener una descripción general del efecto neto de la política en línea y otras políticas de sesión, consulte Servicios a los que pueden acceder los usuarios no autenticados.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:*",
                "logs:*",
                "dynamodb:*",
                "kinesis:*",
                "mobileanalytics:*",
                "s3:*",
                "ses:*",
                "sns:*",
                "sqs:*",
                "lambda:*",
                "machinelearning:*",
                "execute-api:*",
```

```
                "iot:*",
                "gamelift:*",
                "cognito-identity:*",
                "cognito-idp:*",
                "lex:*",
                "polly:*",
                "comprehend:*",
                "translate:*",
                "transcribe:*",
                "rekognition:*",
                "mobiletargeting:*",
                "firehose:*",
                "appsync:*",
                "personalize:*",
                "sagemaker:InvokeEndpoint",
                "cognito-sync:*",
                "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 AWS política de sesiones gestionadas para invitados

Amazon Cognito también aplica una política AWS gestionada como política de sesión a las sesiones de flujo mejorado de los invitados no autenticados. Esta política

limita el ámbito de los permisos de los usuarios no autenticados con la política `AmazonCognitoUnAuthedIdentitiesSessionPolicy`.

También debe conceder este permiso en las políticas que asocie a su rol de IAM no autenticado. Los permisos efectivos de un usuario para una sesión en la que asume un rol son la intersección entre las políticas de IAM que se asignan a su rol y las políticas de sesión. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de AWS Identity and Access Management .

Para obtener una descripción general del efecto neto de esta política AWS administrada y otras políticas de sesión, consulte. Servicios a los que pueden acceder los usuarios no autenticados

La política administrada por `AmazonCognitoUnAuthedIdentitiesSessionPolicy` contiene los permisos siguientes.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "rum:PutRumEvents",
            "polly:*",
            "comprehend:*",
            "translate:*",
            "transcribe:*",
            "rekognition:*",
            "mobiletargeting:*",
            "firehose:*",
            "personalize:*",
            "sagemaker:InvokeEndpoint",
            "geo:GetMap*",
            "geo:SearchPlaceIndex*",
            "geo:GetPlace",
            "geo:CalculateRoute*",
            "geo:*Geofence",
            "geo:*Geofences",
            "geo:*DevicePosition*",
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncryptTo",
            "kms:ReEncryptFrom",
```

```
            "kms:GenerateDataKey",
            "kms:GenerateDataKeyPair",
            "kms:GenerateDataKeyPairWithoutPlaintext",
            "kms:GenerateDataKeyWithoutPlaintext"
        ],
        "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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
```

```
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/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:::amzn-s3-demo-bucket/us-east-1:12345678-1234-1234-1234-123456790ab`. Sin embargo, la identidad no concedería acceso a otros objetos en `arn:aws:s3:::amzn-s3-demo-bucket`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
      "Condition": {"StringLike": {"s3:prefix": ["${cognito-
identity.amazonaws.com:sub}/*"]}}
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/${cognito-
identity.amazonaws.com:sub}/*"]
    }
  ]
}
```

Se logra un modelo de acceso similar con Concesiones de acceso a Amazon S3.

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 las condiciones de la política de IAM para el control de Fine-Grained acceso en la Guía para desarrolladores de Amazon DynamoDB.

JSON

```
{
  "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.

JSON

```
{
    "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.

JSON

```
{
    "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.

JSON

```
{
  "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:

JSON

```
{
  "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 únicamente en los usuarios que hayan iniciado sesión a través de Facebook cambiando la `amr` condición, tal y como se muestra a continuación:

```
"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.

Ante todo, la práctica recomendada más importante al configurar el grupo de identidades es asegurarse de que la aplicación pueda ejecutar su trabajo sin conceder privilegios excesivos o involuntarios. Para evitar errores de configuración de seguridad, antes de lanzar las aplicaciones que desee pasar a producción, revise las recomendaciones que indicamos a continuación.

Temas

- Prácticas recomendadas de configuración de IAM
- Prácticas recomendadas para la configuración del grupo de identidades

Prácticas recomendadas de configuración de IAM

Cuando un usuario invitado o autenticado inicia en la aplicación una sesión que requiere credenciales del grupo de identidades, la aplicación recupera las credenciales temporales de AWS para un rol 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. El usuario obtiene acceso a los recursos de AWS con los permisos asignados a cada rol.

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.

Utilización de condiciones de la política de confianza en roles de IAM

IAM exige que los roles de los grupos de identidades tengan al menos una condición de política de confianza. Esta condición puede, por ejemplo, establecer el ámbito de la función únicamente 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 todos los roles de IAM que confíen en la entidad principal de servicios de los grupos de identidades `cognito-identity.amazonaws.com`.

- `cognito-identity.amazonaws.com:aud`: la notificación aud del token del grupo de identidades debe coincidir con el ID de un grupo de identidades de confianza.
- `cognito-identity.amazonaws.com:amr`: la notificación amr del token del grupo de identidades debe estar autenticada o no autenticada. Con esta condición, puede reservar el acceso a un rol solo a los invitados no autenticados o solo a los usuarios autenticados. Puede añadir aún más precisión al valor de esta condición para restringir el rol a los usuarios de un proveedor específico, por ejemplo, `graph.facebook.com`.

En el ejemplo siguiente de política de confianza de rol se concede acceso a un rol en las siguientes condiciones:

JSON

```
{
  "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 proceder de un grupo de identidades.
- `"cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-cdef-example11111"`: los usuarios deben proceder del grupo de identidades específico `us-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.

Aplicación de permisos de privilegio mínimo

Cuando establezca permisos con políticas de IAM para acceso autenticado o acceso de invitados, conceda solo los permisos específicos necesarios para llevar a cabo tareas concretas o permisos de privilegio mínimo. En el siguiente ejemplo de política de IAM, cuando esta se aplica a un rol, se concede acceso de solo lectura a un único archivo de imagen de un bucket de Amazon S3.

JSON

```
{
  "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 adopte atajos de diseño cuando la aplicación pueda funcionar con los métodos más seguros.

Descripción de los efectos del acceso de invitados

El acceso de invitados no autenticado permite a los usuarios recuperar datos de la Cuenta de AWS que usted posee antes de iniciar sesión. Toda 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, deje el acceso como invitado desactivado y recupere los recursos necesarios solo después de que los usuarios se hayan autenticado. Si su aplicación requiere que se acceda a los recursos antes de iniciar sesión, adopte las siguientes precauciones.

- Familiarícese con las limitaciones automáticas que se imponen a los roles no autenticados.
- Supervise y ajuste los permisos de los roles de IAM no autenticados para que se adapten a las necesidades específicas de la aplicación.
- Conceda acceso a recursos específicos.
- Proteja la política de confianza de su rol de IAM no autenticado predeterminado.
- Active el acceso de invitados solo cuando esté seguro de que concederá los permisos de su rol de IAM a todos los usuarios de Internet.

Utilización de la autenticación mejorada de forma predeterminada

Con la autenticación básica (clásica), Amazon Cognito delega la selección del rol de IAM a su aplicación. Por el contrario, el flujo mejorado utiliza la lógica centralizada del grupo de identidades para determinar el rol de IAM. También proporciona seguridad adicional para las identidades no autenticadas con una política de ámbito reducido que establece un límite máximo para los permisos de IAM. El flujo mejorado es la opción más segura y que exige menos esfuerzo al desarrollador. Para obtener más información sobre estas opciones, consulte Flujo de autenticación de grupos de identidades.

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 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 prácticas recomendadas de IAM a los roles de IAM y a sus permisos.

Utilización de proveedores desarrolladores de forma segura

Las identidades autenticadas por el desarrollador son una característica de los grupos de identidades para las aplicaciones de servidor. La única prueba de autenticación que los grupos de identidades requieren para la autenticación del desarrollador 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 del desarrollador proveedor que se presentan en este flujo de autenticación.

Como práctica recomendada, implemente proveedores desarrolladores únicamente en las siguientes condiciones:

- Para crear responsabilidad respecto al uso de credenciales autenticadas por el desarrollador, diseñe el nombre y los identificadores del proveedor desarrollador para indicar el origen de la autenticación. Por ejemplo: `"Logins" : {"MyCorp provider" : "[provider application ID]"}`.
- Evite las credenciales de usuario de larga duración. Configure el cliente del servidor para que solicite identidades con roles vinculados a servicios, como perfiles de instancia de EC2 o roles de ejecución de Lambda.
- Evite mezclar orígenes de confianza internos y externos en el mismo grupo de identidades. Añada el proveedor desarrollador y los proveedores de inicio de sesión único (SSO) en grupos de identidades diferentes.

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

- Uso de atributos para el control de acceso con grupos de identidades de Amazon Cognito
- Ejemplo de política de uso de atributos para el control de acceso
- Desactivar atributos para el control de acceso (consola)
- Mapeos de proveedores predeterminados

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 concede 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 AWS cuenta
- 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. Concede 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.

> **Important**
>
> Si asigna atributos personalizados de un grupo de usuarios de Amazon Cognito a las etiquetas principales, asegúrese de que los usuarios finales no puedan modificar esos atributos. Un atributo que un usuario pueda modificar no `UpdateUserAttributes` debe

servir como límite de seguridad en sus políticas de IAM. Para evitar que el usuario modifique, realice una de las siguientes acciones:

- Elimine el atributo WriteAttributesde todos los clientes de aplicaciones de su grupo de usuarios, no solo del cliente de aplicaciones que configuró en su grupo de identidades.
- Defina el atributo tal y como `Mutable: false` lo agrega al esquema de su grupo de usuarios.

Todos los clientes de aplicaciones de un grupo de usuarios comparten los atributos de usuario. Si un usuario cambia un atributo a través de cualquier cliente de aplicación, el nuevo valor aparece en los identificadores de ese usuario de cada cliente de aplicación.
Si no especificas un cliente `WriteAttributes` de aplicación, este tiene acceso de escritura a todos los atributos estándar y personalizados. Especifique `WriteAttributes` para cada cliente de aplicación que cree, incluidos los clientes de aplicación que agregue más adelante. Los usuarios no pueden cambiar un atributo inmutable después de crear su perfil de usuario. Sin embargo, pueden establecer su valor al registrarse. Si tu grupo de usuarios permite el registro en modo autoservicio, elimina el atributo de todos los `WriteAttributes` clientes de aplicaciones. A continuación, realice alguna de las operaciones siguientes:

- Establezca el valor al crear usuarios con. `AdminCreateUser`
- Rechaza las solicitudes de registro que contengan valores inesperados. Para obtener más información, consulte the section called “Antes del registro”.

Además, asegúrese de que los valores de los atributos asignados a las etiquetas principales no contengan caracteres que tengan un significado especial en la evaluación de las políticas de IAM, como * (asterisco) o ? (signo de interrogación). En los `Resource` elementos de la política de IAM y en los operadores de `StringLike` condiciones, estos caracteres funcionan como caracteres comodín. Si establece un valor de atributo * y lo asigna a una etiqueta principal, la `${aws:PrincipalTag/`*`tagkey`*`}` evaluación podría coincidir con más recursos de los previstos. Esto puede conceder acceso a los recursos que pertenecen a otros arrendatarios o proyectos. Valide los valores de los atributos en la lógica de su aplicación. Como alternativa, utilice un activador Lambda previo a la generación del token para rechazar o sanear los valores que contienen estos caracteres antes de que la aplicación los pase como etiquetas principales.

Si su aplicación requiere que el usuario pueda escribir el atributo para otros fines, utilice un activador Lambda previo a la generación del token para validar o anular la afirmación antes de que se incluya en el token de ID.

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 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

```
{
            "...": "other claims",
            "sub" : "57e7b692-4f66-480d-98b8-45a6729b4c88",
            "department" : "legal",
```

```
        "clearance" : "confidential",
        "...": "other claims"
    }
```

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

JSON

```
  {
   "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

JSON

```
{
  "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 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" ,	"auth0|5e28d196f8f55a0eaaa95de3", "user123@gmail.com"

Proveedor	Tipo de token	Valores de etiquetas del principal	Ejemplo
		"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"	
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.AGHNIFJQMFSBG3G6XCPVB35ORQAA"
Proveedores estándar de OIDC	Tokens de ID y de acceso	aud (como client_id), sub (como user ID)	"620493171733-eebk7c0hcp5lj3e1tlqp1gntt3k0rncv.apps.googleusercontent.com", "109220063452404746097"
Twitter	Token de acceso	aud (ID de la aplicación; secreto de la aplicación), sub (ID de usuario)	"DfwifTtKEX1FiIBRnOTIR0CFK; Xgj5xb8xIrIVCPjXgLIdkW7fXmwcJJrFvnoK9gwZkLexo1y5z1 «, «1269003884292222976"

Proveedor	Tipo de token	Valores de etiquetas del principal	Ejemplo
DevAuth	Asignación	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:

JSON

```
{
  "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`.

JSON

```
{
    "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`, `StartsWith`o `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 Uso de atributos de usuario.

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.

Para añadir una asignación basada en reglas a un proveedor de identidades en la consola de Amazon Cognito, añada o actualice un IdP y seleccione Elegir un rol con reglas en Selección de rol. Desde allí, puede añadir reglas que asignen las notificaciones del proveedor a los roles de IAM.

Puede configurar un mapeo basado en reglas para los proveedores de identidad en la API AWS CLI o en el `RulesConfiguration` campo del RoleMappingtipo. Puede especificar este campo en el `RoleMappings` parámetro de la SetIdentityPoolRolesAPI.

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 Amazon Cognito ID se representa como un token web JSON (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.
Puede 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 Para obtener AWS las credenciales de los usuarios de la aplicación, utilice el proveedor de credenciales del grupo de identidades Cognito del AWS SDK. Por ejemplo, en el AWS SDK de la versión JavaScript 3, utilice la `fromCognitoIdentityPool` función del paquete. `@aws-sdk/credential-providers` Luego, la identidad se intercambia por credenciales utilizando AWS STS.

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 para JavaScript
- El proveedor de credenciales de Amazon Cognito en la AWS SDK para .NET Guía para desarrolladores
- Especifique las credenciales mediante programación en la guía para desarrolladores AWS SDK para Go
- Proporcione las credenciales temporales en código en la AWS SDK for Java 2.x Guía para desarrolladores
- asuma el RoleWithWebIdentityCredentialProvider proveedor en la Guía para AWS SDK para PHP desarrolladores
- Asumir el rol con el proveedor de identidades web en la documentación de AWS SDK para 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 estos pasos.

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 utilizas la red I/O en el hilo 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 su proveedor no ha establecido un identificador de identidad, se `credentialsProvider.identityId` `nil` devolverán las llamadas. 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 estos 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 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 su proveedor no ha establecido un identificador de identidad, se `credentialsProvider.identityId` `nil` devolverán las llamadas. 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 `fromCognitoIdentityPool`.

Después de configurar un grupo de identidades con sus proveedores de identidad, puede usar la `fromCognitoIdentityPool` función del `@aws-sdk/credential-providers` paquete para autenticar a los usuarios. En lugar de establecer una configuración global`AWS.config.credentials`, asocie el proveedor de credenciales directamente a cada cliente de servicio a través de su `credentials` propiedad. La versión 3 del SDK resuelve el problema del proveedor de forma perezosa la primera vez que el cliente realiza una solicitud, por lo que no existe una llamada independiente para obtener las credenciales. En el siguiente ejemplo, se asocia el proveedor a un: `S3Client`

```
import { S3Client } from "@aws-sdk/client-s3";
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";

// Set the region where your identity pool exists (for example, us-east-1 or eu-west-1)
const region = "REGION";

// Attach the credentials provider to your service client
const client = new S3Client({
  region,
  credentials: fromCognitoIdentityPool({
    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"
    },
    clientConfig: { region },
  }),
});

// Credentials are resolved lazily when the client first makes a request.
// To inspect them directly, call the provider and await the result
// (from inside an async function):
const credentials = await client.config.credentials();
const { accessKeyId, secretAccessKey, sessionToken } = credentials;
```

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
        const region = "us-east-1";
        const client = new S3Client({
          region,
          credentials: fromCognitoIdentityPool({
            identityPoolId: "us-east-1:1699ebc0-7900-4099-b910-2df94f52a030",
            logins: {
              "graph.facebook.com": response.authResponse.accessToken
```

```
            },
            clientConfig: { region },
          }),
        });

        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 su usuario final. Si permite la entrada de usuarios no autenticados, puede recuperarlo inmediatamente. Si autentica a los usuarios, recupérelos después de configurar los identificadores de inicio de sesión. El `fromCognitoIdentityPool` proveedor expone las `identityId` credenciales que resuelve para que puedas leerlas a partir de las credenciales resueltas de un cliente que utilice el proveedor:

```
// Inside an async function:
const { identityId } = await client.config.credentials();
```

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 estos pasos.

El AWS SDK para Unity ahora forma parte de SDK para .NET. Para empezar a utilizar Amazon Cognito en SDK para .NET, consulte el proveedor de credenciales de Amazon Cognito en AWS SDK para .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:

```
string identityId = await credentials.GetIdentityIdAsync();
```

Acceso Servicios de AWS con credenciales temporales

Cuando la autenticación con un grupo de identidades es correcta, se obtiene un conjunto de credenciales de AWS . Con estas credenciales, su aplicación puede realizar solicitudes a AWS los recursos que están protegidos con la autenticación de IAM. Con los distintos AWS SDK que puede añadir a sus aplicaciones para acceder a las operaciones de la API de los grupos de identidades, puede realizar solicitudes de API no autenticadas que generen credenciales temporales. Luego, puedes añadir otros SDK Servicios de AWS a tu cliente y firmar las solicitudes con esas credenciales temporales. Los permisos de IAM otorgados al rol de credenciales temporales deben permitir las operaciones que solicite a otros servicios.

Tras configurar el proveedor de credenciales de Amazon Cognito y recuperar las AWS credenciales, cree un Servicio de AWS cliente. Los siguientes son algunos ejemplos de la documentación del AWS SDK.

AWS Recursos del SDK para crear un cliente

- AWS Configuración del cliente en la Guía para AWS SDK para C++ desarrolladores
- Uso de la AWS SDK para Go V2 con Servicios de AWS la Guía para AWS SDK para 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 para JavaScript desarrolladores
- Creación de clientes en la AWS SDK para 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 el identificador único de los usuarios autenticados y 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 el identificador único de los usuarios autenticados y 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 el identificador único de los usuarios autenticados y 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 el identificador único de los usuarios autenticados y 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 SDK para .NET. Para empezar a utilizar Amazon Cognito en SDK para .NET, consulte el proveedor de credenciales de Amazon Cognito en AWS SDK para .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 el identificador único de los usuarios autenticados y 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 identidades de terceros de grupos de identidades

Con los grupos de identidades de Amazon Cognito, puede integrarse con una variedad de proveedores de identidad externos (IdPs) para proporcionar AWS credenciales temporales mediante la autenticación federada en su aplicación. Al configurar su grupo de identidades para que funcione con estos recursos externos IdPs, puede autorizar el acceso a los AWS recursos de back-end para sus usuarios con la autenticación mediante grupos de usuarios de Amazon Cognito, proveedores de redes sociales, proveedores de OIDC o proveedores de SAML. En esta sección se describen los pasos para configurar e integrar IdPs tu grupo de identidades de Amazon Cognito.

La propiedad `logins` le permite configurar las credenciales recibidas de un proveedor de identidades (IdP). También 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

- Configuración de Facebook como un IdP de grupos de identidades
- Configuración de Login with Amazon como IdP de grupos de identidades
- Configuración de Google como IdP de grupo de identidades
- Configuración de Inicio de sesión con Apple como IdP de grupo de identidades
- Configuración de un proveedor OIDC como IdP de grupo de identidades
- Configuración de un proveedor SAML como IdP de grupo de identidades

Configuración de Facebook como un IdP de grupos de identidades

Los grupos de identidades de Amazon Cognito interactúan con 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:

> **Note**
>
> La federación de grupos de identidades de Amazon Cognito no es compatible con el inicio de sesión limitado de Facebook. Para obtener más información sobre cómo configurar el inicio de sesión con Facebook para iOS sin superar los permisos establecidos para el inicio de sesión limitado, consulte Inicio de sesión con Facebook para iOS: inicio rápido en Meta para desarrolladores.

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 Objective-C las aplicaciones de iOS con el inicio de sesión de Facebook, consulta la Guía de introducción de 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 el flujo de inicio de sesión de Facebook para la web.

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 ha elegido Elegir función con reglas, introduzca la reclamación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que se corresponda 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. 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 seleccione init WithRegionType: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).

> **Note**
>
> La federación de grupos de identidades de Amazon Cognito no es compatible con el inicio de sesión limitado de Facebook. Para obtener más información sobre cómo configurar el inicio de sesión con Facebook para iOS sin superar los permisos establecidos para el inicio de sesión limitado, consulte Inicio de sesión con Facebook para iOS: inicio rápido en Meta para desarrolladores.

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 WithRegionType:identityPoolId:identityProviderManagerinit.

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.

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

FB.login(function (response) {

  // Check if the user logged in successfully.
  if (response.authResponse) {
```

```
    console.log('You are now logged in.');

    // Attach a Cognito identity pool credentials provider to your service client,
    // passing the Facebook access token in the logins map.
    const client = new S3Client({
      region: 'REGION',
      credentials: fromCognitoIdentityPool({
        identityPoolId: 'IDENTITY_POOL_ID',
        logins: {
          'graph.facebook.com': response.authResponse.accessToken
        },
        clientConfig: { region: 'REGION' }
      })
    });

    // The provider resolves and caches AWS credentials the first time the
    // client makes a request. Access AWS resources with client 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**
>
> El proveedor de credenciales resuelve las credenciales de forma AWS lenta la primera vez que el cliente de servicio realiza una solicitud. Para obtener más información sobre el uso de los grupos de identidades de Amazon Cognito con el AWS SDK para JavaScript, consulte la Guía para desarrolladores de AWS SDK for JavaScript v3.

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.

Configuración de Login with Amazon como IdP de grupos de identidades

Los grupos de identidades de Amazon Cognito trabajan con Login with Amazon con el fin de ofrecer autenticación federada para los 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**
>
> 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 un rol con reglas, introduzca la afirmación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que coincida con esta elección de rol y el rol que quiere asignar cuando la asignación de roles 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);
}
```

Usa 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 el método de DidSucceed solicitud de: AMZNAccessTokenDelegate

```
- (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.

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({
   region: 'REGION',
   credentials: fromCognitoIdentityPool({
       identityPoolId: 'IDENTITY_POOL_ID',
       logins: {
           'www.amazon.com': 'Amazon Access Token'
       },
       clientConfig: { region: 'REGION' }
   })
});
```

Configuración de Google como IdP de grupo de identidades

Los grupos de identidades de Amazon Cognito interactúan con Google para ofrecer una autenticación federada a los usuarios de aplicaciones móviles. 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 multicliente de Google, consulta Cross-clientIdentidad.

Configuración de Google

Para activar Google Sign-in para Android, 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 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 acerca de cómo integrar Google en su aplicación web, consulte Authenticate users with Sign in with 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 ha elegido Elegir función con reglas, introduzca la reclamación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que coincida 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. 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 multicliente de Google, consulta Cross-clientIdentidad.

Configuración de Google

Para habilitar Google Sign-in para iOS, crea un proyecto de consola de desarrolladores de Google 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 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 obtener más información sobre cómo integrar Google en tu aplicación para iOS, consulta Google Sign-In 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 has elegido Elegir función con reglas, introduce la reclamación de origen obtenida de la autenticación de tu usuario, el operador con el que quieres comparar la afirmación, el valor que provoca una coincidencia con esta elección de función y la función que quieres asignar cuando la asignación de funciones 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 multicliente de Google, consulta Cross-clientIdentidad.

Configuración de Google

Para habilitar Google Sign-in para iOS, crea un proyecto de consola de desarrolladores de Google 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 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 obtener más información sobre cómo integrar Google en tu aplicación para iOS, consulta Google Sign-In 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 un`id_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 multicliente de Google, consulta Cross-clientIdentidad.

Configuración de Google

Para habilitar Google Sign-in como 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 has elegido Elegir función con reglas, introduce la reclamación de origen obtenida de la autenticación de tu usuario, el operador con el que quieres comparar la afirmación, el valor que provoca una coincidencia con esta elección de función y la función que quieres asignar cuando la asignación de funciones 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:

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

function signinCallback(authResult) {
  if (authResult['status']['signed_in']) {

     // Attach a Cognito identity pool credentials provider to your service client,
     // passing the Google ID token in the logins map.
     const client = new S3Client({
        region: 'REGION',
        credentials: fromCognitoIdentityPool({
```

```
            identityPoolId: 'IDENTITY_POOL_ID',
            logins: {
               'accounts.google.com': authResult['id_token']
            },
            clientConfig: { region: 'REGION' }
         })
      });

      // The provider resolves and caches AWS credentials the first time the
      // client makes a request. Access AWS resources with client here.
   }
}
```

Configuración de Inicio de sesión con Apple como IdP de grupo de identidades

Los grupos de identidades de Amazon Cognito interactúan con Inicio de sesión con Apple con el fin de ofrecer autenticación federada para los 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 identidades (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 obtener la información más actualizada sobre la configuración de inicio de sesión con Apple, consulte Configuración del entorno para iniciar sesión con Apple en la documentación del desarrollador 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. Selecciona Iniciar sesión con Apple y, a continuación, selecciona 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. Selecciona Iniciar sesión con Apple y, a continuación, selecciona 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 ha elegido Elegir función con reglas, introduzca la reclamación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación,

el valor que hace que coincida 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. 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, vea 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 Iniciar sesión con Apple en tu aplicación, consulta Cómo configurar tu página web para Iniciar sesión con Apple en la documentación de Apple.
- Para añadir un botón Iniciar sesión con Apple en una interfaz de usuario Android, siga las instrucciones de Displaying Sign in with Apple buttons on the web 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);
}
```

Usa Iniciar sesión con 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 Iniciar sesión con Apple en tu aplicación, consulta Cómo configurar tu página web para iniciar 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 los botones de inicio de sesión con Apple en la web, 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

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

function signinCallback(authResult) {
     // Attach a Cognito identity pool credentials provider to your service client,
     // passing the Apple ID token in the logins map.
     const client = new S3Client({
        region: 'REGION',
        credentials: fromCognitoIdentityPool({
           identityPoolId: 'IDENTITY_POOL_ID',
           logins: {
              'appleid.apple.com': authResult['id_token']
           },
           clientConfig: { region: 'REGION' }
        })
     });

     // The provider resolves and caches AWS credentials the first time the
     // client makes a request. Access AWS resources with client here.
}
```

Configuración de un proveedor OIDC como IdP de grupo de identidades

OpenID Connect es un estándar abierto de autenticación compatible con varios proveedores de inicio de sesión. Con Amazon Cognito puede vincular identidades con los proveedores 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 ha elegido Elegir función con reglas, introduzca la reclamación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que coincida 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. 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 averiguar 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), Amazon Cognito ofrece un período de gracia de 5 minutos para gestionar cualquier sesgo de 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.

> **Note**
>
> Los grupos de identidades mantienen una memoria caché de la clave de firma del IdP OIDC durante un breve período. Si el proveedor cambia su clave de firma, Amazon Cognito podría devolver un error `NoKeyFound` hasta que se actualice la memoria caché. Si ocurre este error, espere unos diez minutos hasta que su grupo de identidades actualice la clave de firma.

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 }
```

JavaScript

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import { S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({
 region: 'REGION',
 credentials: fromCognitoIdentityPool({
    identityPoolId: 'IDENTITY_POOL_ID',
    logins: {
       'login.provider.com': token
    },
    clientConfig: { region: 'REGION' }
 })
});
```

Configuración de un proveedor SAML como IdP de grupo de identidades

Con los grupos de identidades de Amazon Cognito, puede autenticar a los usuarios con proveedores de identidad (IdPs) mediante 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. Su SAML-supporting IdP especifica las funciones de IAM que pueden asumir sus usuarios. De esta forma, distintos usuarios pueden recibir distintos conjuntos de permisos.

Configuración de un grupo de identidades para un IdP SAML

Los siguientes pasos describen cómo configurar el grupo de identidades para usar un SAML-based IdP.

> **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 identidades SAML del IAM IdPs de su. 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 ha elegido Elegir función con reglas, introduzca la reclamación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que coincida 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. 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, puede usar https://signin.aws.amazon.com/static/saml-metadata.xml. El siguiente paso es 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 identidades determina que la aserción 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 SAML-based IdP. No es necesario especificar un rol autenticado o no autenticado para que el grupo de identidades use un IdP. SAML-based 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 SAML-based IdP, determine la URL en la que el usuario inicia el inicio de sesión. AWS la federación utiliza el inicio de sesión. IdP-initiated 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. Las operaciones de API que pueden aceptar una afirmación de SAML en el `Logins` mapa incluyen GetId, GetCredentialsForIdentityGetOpenIdToken, y. GetOpenIDTokenForDeveloperIdentity 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`.

Developer-authenticated identidades

Amazon Cognito es compatible con las identidades autenticadas por el desarrollador y con la federación de identidades web mediante Configuración de Facebook como un IdP de grupos de identidades, Configuración de Google como IdP de grupo de identidades, Configuración de Login with Amazon como IdP de grupos de identidades y Configuración de Inicio de sesión con Apple como 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 identidades autenticadas por el desarrollador implica la interacción entre el dispositivo del usuario final, su back-end 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. La asignación `Logins` es el nombre de un desarrollador y proveedor del grupo de identidades, como `login.mydevprovider`, emparejado con un identificador personalizado.

Ejemplo:

```
"Logins": {
        "login.mydevprovider": "my developer identifier"
    }
```

Autenticación mejorada

Llama a la operación de la GetCredentialsForIdentityAPI con un `Logins` mapa en el que la clave sea el emisor del token devuelto y el valor sea el propio token. `GetOpenIdTokenForDeveloperIdentity`

El emisor (y, por tanto, la clave de inicio de sesión requerida) depende de la región del grupo de identidades. Consulta Emisor del token por región la lista completa de regiones y sus emisores de tokens.

Ejemplo de un grupo de identidades en una región que utiliza el emisor global:

```
"Logins": {
        "cognito-identity.amazonaws.com": "eyJra12345EXAMPLE"
    }
```

Ejemplo de un grupo de identidades en una región más nueva (como eu-south-2):

```
"Logins": {
        "cognito-identity.eu-south-2.amazonaws.com": "eyJra12345EXAMPLE"
    }
```

> **Important**
>
> La clave del mapa de inicios de sesión debe coincidir exactamente con la afirmación `iss` (del emisor) que figura en el token. Si no coinciden, `GetCredentialsForIdentity` devuelve:

```
NotAuthorizedException: Invalid login token. Issuer doesn't match
providerName
```

> **Tip**
>
> Para determinar la clave de inicio de sesión correcta mediante programación, decodifique el token (un JWT) y lea el valor declarado. `iss` Usa ese valor como clave de mapa de inicios de sesión.

`GetCredentialsForIdentity` con identidades autenticadas por el desarrollador, devuelve credenciales temporales para el rol autenticado predeterminado del grupo de identidades.

Autenticación básica

Llame a la operación de la AssumeRoleWithWebIdentityAPI y solicite la `RoleArn` de cualquier función de IAM que tenga definida una relación de confianza adecuada. Defina el valor de `WebIdentityToken` en el token obtenido de `GetOpenIdTokenForDeveloperIdentity`.

Para obtener más información acerca del flujo de autenticación de las identidades autenticadas por el desarrollador y en qué se diferencian de las identidades de proveedores externos, consulte Flujo de autenticación de grupos de identidades.

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.

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", Regions.USEAST1);
CognitoCachingCredentialsProvider credentialsProvider = new
 CognitoCachingCredentialsProvider( context, developerProvider, Regions.USEAST1);
```

> **Note**
>
> Las versiones anteriores del AWS SDK para Android (2.x y anteriores) se utilizan `cognito-identity.amazonaws.com` como nombre de proveedor interno para las identidades autenticadas por el desarrollador. Si tu grupo de identidades se encuentra en una región más reciente que usa un emisor regionalizado, actualiza a una versión actual del SDK. Para obtener más información, consulte Formato de emisor de fichas regional.

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];
    }
}
```

> **Note**
>
> Las versiones anteriores del AWS SDK para iOS (2.x y anteriores) se utilizan `cognito-identity.amazonaws.com` como nombre de proveedor interno para las identidades autenticadas por el desarrollador. Si su grupo de identidades se encuentra en una región más reciente que utiliza un emisor regionalizado, actualícelo a una versión actual del SDK. Para obtener más información, consulte Formato de emisor de fichas regional.

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 obtengas un identificador de identidad y un token de sesión de tu servidor, los pasarás al `fromCognitoIdentity` proveedor desde allí. `@aws-sdk/credential-providers` La clave de inicio de sesión debe coincidir con la `iss` afirmación del token. Para los grupos de identidades en las regiones que utilizan el emisor global, utilice. `cognito-identity.amazonaws.com` Para grupos de identidades en otras regiones, utilice`cognito-identity.region.amazonaws.com`. Para obtener más información, consulte Formato de emisor de fichas regional.

```
import { fromCognitoIdentity } from "@aws-sdk/credential-providers";

const token = 'TOKEN_RETURNED_FROM_YOUR_PROVIDER';
const region = 'IDENTITY_POOL_REGION';

// Extract the issuer from the token
const payload = token.split('.')[1];
const claims = JSON.parse(Buffer.from(payload, 'base64url').toString());
const providerName = claims.iss;

// Your backend already established the identity, so use fromCognitoIdentity
// (which calls GetCredentialsForIdentity directly for the identity ID you pass)
// rather than fromCognitoIdentityPool (which would call GetId and resolve a new
 identity).
const credentials = fromCognitoIdentity({
   identityId: 'IDENTITY_ID_RETURNED_FROM_YOUR_PROVIDER',
   logins: {
      [providerName]: token
   },
   clientConfig: { region }
});

// Attach the provider to each service client that needs credentials.
const client = new SomeServiceClient({ region, credentials });
```

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)

Al llamar, se obtiene una ficha GetOpenIdTokenForDeveloperIdentity. Esta API debe invocarse desde tu back-end 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 tienes algún problema o pregunta, publica en nuestros foros.

Formato de emisor de fichas regional

El token de OpenID Connect devuelto por `GetOpenIdTokenForDeveloperIdentity` contiene una notificación `iss` (del emisor). El formato de esta afirmación depende de la AWS región del grupo de identidades.

Cuando llames `GetCredentialsForIdentity` (flujo mejorado), la clave del mapa de inicios de sesión debe coincidir exactamente con la `iss` afirmación del token.

Emisor del token por región

El valor del emisor del token depende de la región de tu grupo de identidades.

- Regiones que utilizan el emisor global (us-east-1, us-east-2, us-west-1, us-west-2, ap-northeast-1, ap-northeast-2, ap-northeast-2, ap-northeast-3, ap-south-1, ap-southeast-1 ap-southeast-2, ca-central-1, eu-central-1, eu-north-1, eu-west-1, eu-west-2, eu-west-3, sa-east-1): `cognito-identity.amazonaws.com`
- Las regiones de China (cn-north-1): `cognito-identity.cn-north-1.amazonaws.com.cn`
- Todas las demás regiones comerciales: AWS `cognito-identity.`*`region`*`.amazonaws.com`

Determinar la clave de inicio de sesión mediante programación

En lugar de codificar la clave de inicio de sesión, puedes extraerla del token:

Java

```
import java.util.Base64;
```

```
import org.json.JSONObject;

String[] parts = token.split("\\.");
String payload = new String(Base64.getUrlDecoder().decode(parts[1]));
String issuer = new JSONObject(payload).getString("iss");
// Use 'issuer' as the logins map key in GetCredentialsForIdentity
```

Python

```
import base64, json

payload = token.split('.')[1]
# Add padding if needed
payload += '=' * (4 - len(payload) % 4)
claims = json.loads(base64.urlsafe_b64decode(payload))
issuer = claims['iss']
# Use 'issuer' as the logins map key
```

JavaScript

```
const payload = token.split('.')[1];
const claims = JSON.parse(Buffer.from(payload, 'base64url').toString());
const issuer = claims.iss;
// Use 'issuer' as the logins map key
```

Consideraciones sobre la migración

Si va a migrar un grupo de identidades a una región que utiliza un emisor regionalizado, tenga en cuenta lo siguiente:

1. Los clientes que codifiquen `cognito-identity.amazonaws.com` como clave de inicio de sesión fallarán en el conjunto de identidades de la nueva región.
2. El código de cliente debe actualizarse para utilizar el emisor regionalizado (`cognito-identity.`*`new-region`*`.amazonaws.com`) o para leer de forma dinámica la `iss` reclamación del token (recomendado).
3. Las versiones anteriores de los SDK AWS móviles codifican de forma rígida el emisor global. Esto incluye el AWS SDK para Android 2.x y versiones anteriores y el AWS SDK para iOS 2.x y versiones anteriores. Si su grupo de identidades usa un emisor regionalizado, actualícelo a una versión actual del SDK.

Conexión con una identidad social existente

Todas las vinculaciones de proveedores cuando utilices identidades autenticadas por el desarrollador deben realizarse desde tu back-end. 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 que esto sea posible, cuando llames a tu servidor desde el SDK de tu cliente para autenticar a tu usuario final, también debes pasarle el token del proveedor de redes sociales 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 otras identidades, la aplicación móvil interactúa directamente con Amazon Cognito en lugar de ponerse en contacto 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 desarrollador, póngase en contacto con su equipo de administración. De lo contrario, llama al IdentityId método get 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;
   }
}
```

Del mismo modo, el `getIdentityId` método tiene dos flujos en función del contenido del mapa 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 borra tanto las credenciales como la identidad y obliga 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 borra tanto las credenciales como la identidad y obliga 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 otras identidades, la aplicación móvil interactúa directamente con Amazon Cognito en lugar de ponerse en contacto 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 consiste en ampliar el proveedor de identidad desde AmazonCognitoEnhancedIdentityProvide y llamar al RefreshAsync método principal en lugar del propio AbstractCognitoIdentityProvider, en caso de que el usuario no esté autenticado con su propio servidor. Si el usuario está autenticado, puede utilizar el mismo flujo explicado anteriormente.

Transición de usuarios sin autenticar a usuarios autenticados

Los grupos de identidades de Amazon Cognito admiten usuarios no autenticados y autenticados. Los usuarios no autenticados reciben acceso a tus AWS recursos aunque no hayan iniciado sesión con ninguno de tus proveedores de identidad ()IdPs. Este grado de acceso es útil para mostrar el contenido a los usuarios antes de que inicien sesión. Cada usuario no autenticado tiene una identidad única en el grupo de identidades, aunque no haya iniciado sesión ni se haya autenticado individualmente.

En esta sección se describe el caso en el que el usuario decide pasar de iniciar sesión con una identidad no autenticada a usar una identidad autenticada.

Android

Los usuarios pueden iniciar sesión en tu aplicación como invitados no autenticados. En algún momento, es posible que decidan iniciar sesión con una de las opciones compatibles. IdPs 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 tu aplicación como invitados no autenticados. En algún momento, es posible que decidan iniciar sesión con una de las opciones compatibles. IdPs 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 tu aplicación como invitados no autenticados. En algún momento, es posible que decidan iniciar sesión con una de las opciones compatibles. IdPs 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, debe crear un proveedor de `fromCognitoIdentityPool` credenciales sin una `logins` propiedad y adjuntarlo a sus clientes de servicio. En este caso, la configuración predeterminada podría tener el siguiente aspecto:

```
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";

// Create a credentials provider without a logins map for the unauthenticated role.
const region = "us-east-1";
const credentials = fromCognitoIdentityPool({
    identityPoolId: "us-east-1:1699ebc0-7900-4099-b910-2df94f52a030",
    clientConfig: { region }
});

// Attach the provider to each service client that needs credentials.
const client = new SomeServiceClient({ region, credentials });
```

Cambie a usuario autenticado

Cuando un usuario no autenticado inicia sesión en un IdP y tú tienes un token, puedes cambiar el usuario de no autenticado a autenticado llamando a una función personalizada que crea un nuevo `fromCognitoIdentityPool` proveedor con el mapa que contiene el token: `logins`

```
// Called when an identity provider has a token for a logged in user
function userLoggedIn(providerName, token) {
    // Re-create the provider with a logins map to switch to the authenticated role.
    const credentials = fromCognitoIdentityPool({
        identityPoolId: "us-east-1:1699ebc0-7900-4099-b910-2df94f52a030",
        logins: {
            [providerName]: token
        },
        clientConfig: { region }
    });

    // Re-create any service clients so they pick up the new credentials.
    // SomeServiceClient is a placeholder: import the client for the service you're
 calling
```

```
    // (for example, S3Client from @aws-sdk/client-s3).
    const client = new SomeServiceClient({ region, credentials });
}
```

Como el `fromCognitoIdentityPool` proveedor está vinculado directamente a cada cliente de servicio, cambiar del rol no autenticado al autenticado implica crear un nuevo proveedor con el `logins` mapa y pasarlo a cualquier cliente de servicio que cree posteriormente. A diferencia del AWS SDK de la JavaScript versión 2, el SDK de la versión 3 no utiliza un objeto de configuración global, por lo que no es necesario restablecer ninguna credencial compartida.

Para obtener más información sobre el `fromCognitoIdentityPool` proveedor, consulte el AWS SDK para proveedores de credenciales de la JavaScript versión 3.

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. Su grupo de identidades puede incorporar identidades de los siguientes tipos de servicios de autenticación:

- 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.
- Los usuarios se autenticaron mediante su propio proceso de autenticación existente

Una vez que los usuarios se autentican con su proveedor y presentan la autorización a un grupo de identidades, obtienen credenciales temporales AWS . Las credenciales de los usuarios tienen permisos que se definen para acceder a otros Servicios de AWS.

Temas

- Creación de un grupo de identidades en Amazon Cognito
- Configurar un SDK
- Integración de los proveedores de identidad
- Obtención de credenciales
- Ejemplo de aplicación para grupos de identidades

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. El siguiente procedimiento es una guía general para crear un nuevo grupo de identidades en la consola. También puede ir directamente a la consola y seguir la experiencia guiada y el contenido de la ayuda integrada.

Para crear un grupo de identidades nuevo en la consola

1. Inicie sesión en la consola de Amazon Cognito y seleccione Grupos de identidades. A fin de asignar permisos a la entidad principal de IAM para que pueda crear y gestionar los recursos de Amazon Cognito, consulte AWS políticas gestionadas para Amazon Cognito. La política de `AmazonCognitoPowerUser` es suficiente para crear 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 asigna a su nueva función 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 función con reglas, introduzca la afirmación de origen obtenida de la autenticación de su usuario, el operador con el que quiere comparar la afirmación, el valor que hace que coincida 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. 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, puedes omitir los roles que seleccionaste para ti IdPs y llamar AssumeRoleWithWebIdentitydirectamente. Para obtener más información, consulte Flujo de autenticación de grupos de identidades.

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 para Java, el o el. SDK para .NET Para obtener más información, consulte los siguientes temas.

- Cómo configurar el SDK de JavaScript la Guía para AWS SDK para JavaScript desarrolladores
- Documentación de Amplify en el Amplify Dev Center
- Proveedor de credenciales de Amazon Cognito en la Guía para desarrolladores de SDK para .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. También es compatible con esta funciónDeveloper-authenticated identidades, que te permite registrar y autenticar a los usuarios mediante tu propio proceso de autenticación back-end.

Si desea obtener más información sobre el uso de un grupo de usuarios de Amazon Cognito para crear su propio directorio de usuarios, consulte Autenticación de usuarios con grupos de usuarios de Amazon Cognito y Acceso Servicios de AWS usar un grupo de identidades después de iniciar sesión.

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

Para obtener más información acerca de la integración de su propio proceso de autenticación de backend, consulte Developer-authenticated 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 back-end interno AWS o externo AWS a través de Amazon API Gateway. Consulte Obtención de credenciales.

Ejemplo de aplicación para grupos de identidades

El caso de uso más común de los grupos de identidades de Amazon Cognito es federar usuarios de varios sistemas de inicio de sesión y entregar credenciales temporales de acceso limitado directamente AWS al cliente. Esto elimina la necesidad de crear un agente de credenciales para obtener permisos de acceso a sus recursos. AWS Por ejemplo, puede que tenga que permitir que los usuarios inicien sesión con sus cuentas de redes sociales y accedan a los activos de la aplicación desde Amazon S3 para su aplicación móvil. Los grupos de identidades también proporcionan credenciales a los usuarios que inician sesión con grupos de usuarios.

En este tutorial, creará una aplicación web en la que podrá obtener credenciales temporales autenticadas y de invitado en los flujos de autenticación básica y mejorada con proveedores de identidad compatibles (IdPs) en los grupos de identidades. Si ya tiene experiencia en el desarrollo web, descargue la aplicación de ejemplo desde GitHub.

Descarga la aplicación de ejemplo desde GitHub

En este ejemplo de aplicación, se muestran las siguientes capacidades de los grupos de identidades de Amazon Cognito:

Flujos de autenticación en grupos de identidades

- Flujo de autenticación mejorado con desgloses detallados de las solicitudes de API
- Flujo de autenticación básico con desgloses detallados de las solicitudes de API

Implementación del acceso como invitado (no autenticado)

- Proporcione Servicio de AWS acceso limitado sin necesidad de iniciar sesión

Integración con proveedores de identidad compatibles

- Redes sociales IdPs (Facebook, Amazon, Twitter, Apple y Google) para el acceso de los consumidores
- Enterprise IdPs (a través de OpenID Connect o SAML) para usuarios corporativos
- Grupos de usuarios de Amazon Cognito

AWS administración de credenciales

- Intercambio de los tokens de los proveedores de identidad por credenciales de AWS temporales
- Uso de credenciales temporales para acceder a los AWS servicios de forma segura

Tras configurar la aplicación en el servidor web de desarrollo y acceder a ella desde un navegador, aparecen las siguientes opciones.



Temas

- Requisitos previos
- Configuración del proveedor de autenticación
- Implementación de la aplicación de demostración
- Explore los métodos de autenticación de su grupo de identidades
- Siguientes pasos

Requisitos previos

Antes de comenzar, necesitará configurar los siguientes recursos:

- Una AWS cuenta con acceso a Amazon Cognito. Si no tiene una AWS cuenta, siga las instrucciones que se indican enIntroducción al AWS.
- Python 3.11 o posterior instalado en su máquina de desarrollo.
- GitHub acceso.
- AWS credenciales configuradas con permisos para realizar solicitudes autenticadas a las API de Amazon Cognito. Estas credenciales son necesarias para la autenticación de los desarrolladores.

Para obtener más información sobre la implementación de la federación de AWS credenciales y grupos de identidades en su SDK específico, consulte. the section called “Obtención de credenciales”

Configuración del proveedor de autenticación

Para obtener los mejores resultados con esta aplicación, configure e integre uno o más proveedores de identidades de terceros (IdPs) o grupos de usuarios de Amazon Cognito con su grupo de identidades de Amazon Cognito. Tras completar los requisitos previos y antes de ejecutar esta aplicación de demostración, elija los proveedores de identidad que desee configurar. La consola de Amazon Cognito le guía por el proceso de configuración de proveedores y grupos de identidades.

Grupos de usuarios de Amazon Cognito

- the section called “Autenticación”
- the section called “Clientes de aplicaciones”

Proveedores de identidades de redes sociales

- Google: the section called “Google”
- Facebook: the section called “Facebook”
- Amazon: the section called “Login with Amazon”

Proveedores de OpenID Connect (OIDC)

- the section called “Proveedores de Open ID Connect”

Proveedores SAML

- the section called “Proveedores de identidades SAML”

> **Note**
>
> Para esta aplicación de demostración, no tiene que configurar todos los proveedores de identidades compatibles. Puede empezar con uno que coincida con su caso de uso. Cada enlace proporciona instrucciones de configuración detalladas.

Implementación de la aplicación de demostración

Clone el repositorio

1. Abra una ventana de terminal.

2. Clone el repositorio `aws-doc-sdk-examples` o recupera de otro modo esta carpeta en el repositorio.

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

3. Desplácese hasta el directorio del proyecto .

```
cd python/example_code/cognito/scenarios/identity_pools_example_demo/web
```

Creación de un grupo de identidades .

A fin de crear un grupo de identidades de Amazon Cognito para su aplicación, siga las instrucciones en the section called “Configuración de grupos de identidades”.

Cómo configurar un grupo de identidades para la aplicación de demostración

1. Abra la consola de Amazon Cognito.
2. En el panel de navegación de la izquierda, elija Grupos de identidades. Seleccione un grupo de identidades existente o cree uno nuevo.
3. En Acceso de usuario, habilite el Acceso autenticado y el Acceso de invitado. Configure un rol de IAM nuevo o existente y asígnele los permisos que desee conceder a cada tipo de usuario.
4. En Acceso de usuario, configure los proveedores de identidad que desee configurar.
5. En Propiedades del grupo de identidades, habilite la Autenticación básica (clásica).
6. Mantenga el navegador abierto con la consola de su grupo de identidades. Utilizará el ID del grupo de identidades y otros datos de ajuste en la configuración de la aplicación.

Configuración y ejecución de la aplicación

Los pasos siguientes le guiarán a través de la configuración inicial de su aplicación de demostración.

Cómo configurar la aplicación de demostración

1. Abra una línea de comandos en `python/example_code/cognito/scenarios/identity_pools_example_demo/web` en su clon de `aws-doc-sdk-examples`.
2. Cree un archivo `.env` copiando el archivo de entorno de ejemplo.

```
cp .env.example .env
```

3. Abra el archivo `.env` en un editor de texto. Sustituya el resto de los valores de ejemplo por sus propios valores.
4. Instale dependencias de backend.

```
pip install -r requirements.txt
```

5. Inicie el servidor backend:

```
cd backend
python oauth_server.py
```

6. Abra una nueva ventana de terminal, navegue hasta el directorio del proyecto e inicie el servidor del frontend:

```
cd frontend
python -m http.server 8001
```

7. Abre la aplicación en tu navegador en: 8001http://localhost. Su navegador muestra la interfaz de la aplicación de demostración, lista para probar la autenticación de los grupos de identidades.

Explore los métodos de autenticación de su grupo de identidades

Esta sección le guía a través de los flujos de autenticación básicos y mejorados mediante la aplicación de demostración de grupos de identidades de Amazon Cognito. Con esta demostración, aprenderá cómo funcionan los grupos de identidades con varios proveedores de identidades para proporcionar AWS credenciales temporales a los usuarios de su aplicación.

En la sección Demostración interactiva de la aplicación de ejemplo, primero elegirá entre dos tipos de acceso compatibles con los grupos de identidades.

Acceso no autenticado (invitado)

Proporcione AWS credenciales a los usuarios que aún no se hayan autenticado.

Acceso autenticado

Cambie los tokens de los proveedores de identidad por AWS credenciales con una gama completa de permisos disponibles. Elija un proveedor de identidad de entre los que configuró en su archivo `.env`.

Acceso no autenticado (invitado)

En este paso, se muestra cómo obtener AWS credenciales temporales para usuarios no autenticados (invitados) mediante la función de acceso como invitado de su grupo de identidades. En la aplicación de demostración, probará los flujos básicos y mejorados para ver cómo los grupos de identidades emiten credenciales sin necesidad de que el usuario inicie sesión. El acceso como invitado utiliza la misma secuencia de API que el acceso autenticado, pero sin proporcionar los identificadores de los proveedores de identidad (como los tokens OAuth de Google, Facebook o las aserciones de SAML de los proveedores empresariales).

Siga leyendo si busca información sobre cómo proporcionar acceso limitado en AWS a los usuarios sin requerir autenticación. Tras implementar el acceso como invitado, aprenderás a proporcionar AWS credenciales de forma segura a los usuarios anónimos y a entender las diferencias entre los dos flujos de autenticación.

> **Important**
>
> El acceso no autenticado puede emitir credenciales a cualquier persona con acceso a Internet, por lo que es mejor utilizarlo para AWS recursos que requieren una seguridad mínima, como las API públicas y los recursos gráficos. Antes de continuar con este paso, compruebe si ha configurado su grupo de identidades con el acceso como invitado habilitado y asegúrese de que existen las políticas de IAM adecuadas para limitar los permisos.

Guest access with enhanced flow

El flujo mejorado es un enfoque simplificado para obtener credenciales de AWS para los usuarios no autenticados con dos solicitudes de API.

Cómo probar el acceso de invitados con el flujo mejorado

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Seleccione la pestaña Acceso de invitado.
3. Seleccione la pestaña Flujo mejorado.
4. Seleccione Probar acceso de invitado.
5. La aplicación obtiene AWS credenciales temporales de tus grupos de identidades sin necesidad de solicitar más autenticación.

6. Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

   a. Botón Ver solo credenciales: seleccione este botón si desea ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-07T00:58:21-07:00"
     }
   }
   ```

   b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

   - Solicitud de API `GetId()` con su `identityPoolId`. No se requieren tokens de autenticación para el acceso de invitados

     ```
     {
       "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
     }
     ```

     Si es válido, busca o crea y devuelve el `IdentityID` del usuario. Un ejemplo de respuesta:

     ```
     {
       "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
     }
     ```

   - `GetCredentialsForIdentity()` con el `identityPoolId` devuelto.

     ```
     POST GetCredentialsForIdentity
     {
       "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
     }
     ```

Cognito valida el acceso de invitado, asume internamente el rol de no autenticado AWS STS y devuelve una credencial de AWS temporal. (No hay autenticación de IAM en esta llamada; la confianza en el rol debe permitir `cognito-identity-amazonzaws.com`).

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-07T00:58:21-07:00"
  }
}
```

Guest access with basic flow

El flujo básico proporciona un control pormenorizado del proceso de autenticación, con solicitudes de API independientes para la recuperación de la identidad y la generación de credenciales.

Cómo probar el acceso de invitados con el flujo básico

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Seleccione la pestaña Acceso de invitado.
3. Seleccione la pestaña Flujo básico.
4. Seleccione Probar acceso de invitado.
5. La aplicación obtiene AWS credenciales temporales de sus grupos de identidades sin necesidad de solicitar más autenticación.
6. Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

   a. Botón Ver solo credenciales: seleccione este botón si desea ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
```

```
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

- Solicitud de API `GetId()` con su ID de grupo de identidades. No se requieren tokens de autenticación para el acceso de invitados.

```
POST GetId
{
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Si es válido, busca o crea y devuelve el `IdentityID` del usuario. Un ejemplo de respuesta:

```
{
   "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetOpenIdToken()` con el `IdentityID` devuelto y el mismo mapa `Logins`

```
POST GetOpenIdToken
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

Qué ocurre en este paso: Amazon Cognito emite un token de identidad web OpenID Connect de corta duración desde cognito-identity.amazonaws.com que representa a este `IdentityId`. El token incluye las afirmaciones del OIDC que AWS STS evalúan,

como aud (el ID de tu grupo de identidades) y amr (autenticado o no autenticado). La política de confianza de su rol de IAM debe exigir esas reclamaciones.

- `AssumeRoleWithWebIdentity()`- Tu aplicación llama AWS STS directamente para cambiar el token OpenID de Amazon Cognito por credenciales temporales AWS

```
POST sts:AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/
Cognito_IdentityPoolUnauth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......"
}
```

Respuesta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "FwoGZXIvYXdzEEXAMPLE......"
  }
}
```

Qué ocurre en este paso: una vez validadas, devuelve las credenciales temporales de AWS

Uso de credenciales temporales

Estas credenciales temporales funcionan como AWS credenciales estándar, pero con permisos limitados definidos por la función de IAM no autenticada del grupo de identidades. Puede utilizarlas con cualquier AWS SDK o. AWS CLI Para obtener más información sobre la configuración de AWS los SDK con credenciales, consulte los proveedores de credenciales estandarizados en la Guía de referencia de AWS SDK y herramientas.

Los ejemplos que aparecen a continuación no son una lista completa, pero muestran las formas habituales en las que la característica de invitado de un grupo de identidades puede mejorar la experiencia del usuario.

Contenido público de solo lectura

Los siguientes ejemplos configuran los proveedores de credenciales para el acceso limitado a Amazon S3 como usuario invitado.

Python

```
# Example: Using credentials with boto3
import boto3

# Configure client with temporary credentials
s3_client = boto3.client(
    's3',
    aws_access_key_id='AKIAIOSFODNN7EXAMPLE',
    aws_secret_access_key='wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
    aws_session_token='IQoJb3JpZ2luX2VjEEXAMPLE......'
)

# Make API requests within IAM role permissions
response = s3_client.list_objects_v2(Bucket='my-public-bucket')

# Access public content
for obj in response.get('Contents', []):
    print(f"File: {obj['Key']}, Size: {obj['Size']} bytes")
```

JavaScript

```
// Example: Accessing public content
import { S3Client, GetObjectCommand } from "@aws-sdk/client-s3";

const s3Client = new S3Client({
    region: "us-east-1",
    credentials: {
        accessKeyId: 'AKIAIOSFODNN7EXAMPLE',
        secretAccessKey: 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
        sessionToken: 'IQoJb3JpZ2luX2VjEEXAMPLE......'
    }
});

// Access public images or documents
const response = await s3Client.send(new GetObjectCommand({
    Bucket: 'my-public-content',
    Key: 'product-catalog.pdf'
```

```
}));
```

características de "» Try-before-login

Los ejemplos siguientes utilizan el acceso de solo lectura a Amazon DynamoDB como usuario invitado.

Python

```
# Example: Limited app functionality for trial users
import boto3

dynamodb = boto3.client(
    'dynamodb',
    aws_access_key_id='AKIAIOSFODNN7EXAMPLE',
    aws_secret_access_key='wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
    aws_session_token='IQoJb3JpZ2luX2VjEEXAMPLE......'
)

# Allow guest users to view sample data (limited to 5 items)
response = dynamodb.scan(TableName='SampleProducts', Limit=5)
```

JavaScript

```
// Example: Limited app functionality for trial users
import { DynamoDBClient, ScanCommand } from "@aws-sdk/client-dynamodb";

const dynamodbClient = new DynamoDBClient({
    region: "us-east-1",
    credentials: {
        accessKeyId: 'AKIAIOSFODNN7EXAMPLE',
        secretAccessKey: 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
        sessionToken: 'IQoJb3JpZ2luX2VjEEXAMPLE......'
    }
});

// Allow guest users to view sample data (limited to 5 items)
const response = await dynamodbClient.send(new ScanCommand({
    TableName: 'SampleProducts',
    Limit: 5
}));
```

Autenticación de proveedores de identidad sociales

En este paso, se analiza el flujo general de uso de proveedores de identidades sociales con los grupos de identidades de Amazon Cognito. La autenticación social proporciona una experiencia de inicio de sesión familiar y, al mismo tiempo, mantiene la seguridad mediante la administración de identidades federadas. Puedes iniciar sesión desde un proveedor de identidad social (IdP) como Google, Facebook y Amazon, y luego cambiar ese token de IdP por credenciales temporales. AWS Los grupos de identidades también admiten la integración de Twitter y Apple, pero no en la aplicación de ejemplo.

El grupo de identidades en sí no es un directorio de usuarios. No almacena contraseñas ni campos de perfil. En lugar de ello, confía en una entidad externa IdPs para autenticar al usuario y se centra en autorizar a ese usuario ya autenticado a llamar directamente a los AWS servicios mediante la venta de credenciales para las funciones de IAM.

Social identity provider with enhanced flow

En esta sección, se muestra cómo puede utilizar un proveedor de identidad social para iniciar sesión en un usuario y, mediante el flujo mejorado, cambiar el token del proveedor en un grupo de identidades de Amazon Cognito por credenciales temporales para solicitar recursos de AWS .

Uso del inicio de sesión por redes sociales con el flujo mejorado en la aplicación de ejemplo

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Elija la pestaña Acceso autenticado.
3. Seleccione la pestaña Flujo mejorado.
4. Elija un proveedor social compatible que haya configurado, como Iniciar sesión con Google, Iniciar sesión con Facebook o Iniciar sesión con Amazon.
5. Inicie sesión y acepte compartir los datos del usuario con la aplicación.
6. El proveedor redirige de nuevo al URI de redireccionamiento de la aplicación
7. La aplicación envía el token del proveedor a tu grupo de identidades y recupera las credenciales temporales AWS
8. La aplicación muestra el panel de resultados en la interfaz web.

   Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

a. Botón Ver solo credenciales: selecciona este botón si quieres ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

- La aplicación inicia sesión en el usuario con un IdP social y obtiene el token del proveedor. Los grupos de identidades aceptan los siguientes artefactos de los proveedores:

Proveedor de identidades	Clave de proveedor de Cognito	Finalidad
Google	`accounts.google.com`	Tokens de OAuth 2.0 de Google Sign-In
Facebook	`graph.facebook.com`	Tokens de acceso de Facebook Login
Amazon	`www.amazon.com`	Tokens OAuth desde Login with Amazon

Tras la autenticación correcta con el proveedor social, su aplicación recibe una respuesta de OAuth que contiene el token de acceso y otros detalles de autenticación:

```
{
   "access_token": "ya29.A0AS3H6NEXAMPLE......",
   "expires_in": 3599,
   "scope": "openid https://www.examplesocial....",
```

```
   "token_type": "Bearer",
   "id_token": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
}
```

- La solicitud de API `GetId()` con el ID de su grupo de identidades y un mapa `Logins` que contiene el token de su proveedor social

```
POST GetId
{
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "accounts.google.com": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetCredentialsForIdentity()` con el `IdentityID` devuelto y el mismo mapa `Logins`

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "accounts.google.com": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
  }
}
```

Respuesta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-07T00:58:21-07:00"
  },
```

```
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Qué ocurrió: Amazon Cognito validó el token con el proveedor configurado, eligió un rol de IAM en función de la configuración del proveedor y llamó AWS STS en su nombre. A continuación, su grupo de identidades devolvió credenciales temporales.

Social identity provider with basic flow

En esta sección se muestra cómo puede utilizar un proveedor de identidad social para iniciar sesión en un usuario y, mediante el flujo básico, cambiar el token del proveedor en un grupo de identidades de Amazon Cognito por credenciales temporales para llamar AWS a los servicios.

Uso del inicio de sesión por redes sociales con el flujo básico en la aplicación de ejemplo

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Elija la pestaña Acceso autenticado.
3. Seleccione la pestaña Flujo básico.
4. Elija un proveedor social compatible que haya configurado, como Iniciar sesión con Google, Iniciar sesión con Facebook o Iniciar sesión con Amazon.
5. Inicie sesión y acepte compartir los datos del usuario con la aplicación.
6. El proveedor redirige de nuevo al URI de redireccionamiento de la aplicación
7. La aplicación envía el token del proveedor a su grupo de identidades y recupera las credenciales temporales AWS
8. La aplicación muestra el panel de resultados en la interfaz web.

   Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

   a. Botón Ver solo credenciales: selecciona este botón si quieres ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
```

```
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

- La aplicación inicia sesión en el usuario con un IdP social y obtiene el token del proveedor. Los grupos de identidades aceptan los siguientes artefactos de los proveedores:

Proveedor de identidades	Clave de proveedor de Cognito	Finalidad
Google	`accounts.google.com`	Tokens de OAuth 2.0 de Google Sign-In
Facebook	`graph.facebook.com`	Tokens de acceso de Facebook Login
Amazon	`www.amazon.com`	Tokens OAuth desde Login with Amazon

Tras la autenticación correcta con el proveedor social, su aplicación recibe una respuesta de OAuth que contiene el token de acceso y otros detalles de autenticación:

```
{
   "access_token": "ya29.A0AS3H6NEXAMPLE......",
   "expires_in": 3599,
   "scope": "openid https://www.examplesocial....",
   "token_type": "Bearer",
   "id_token": "eyJhbGciOiJSUzI1NiIsEXAMPLE......"
}
```

- La solicitud de API `GetId()` con el ID de su grupo de identidades y un mapa `Logins` que contiene el token de su proveedor social

```
POST GetId
{
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
  "Logins": {
    "accounts.google.com": "token..."
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetOpenIdToken()` con el `IdentityID` devuelto y el mismo mapa Logins

```
POST GetOpenIdToken
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "accounts.google.com": "token..."
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

- `AssumeRoleWithWebIdentity()` con el token OpenID

```
POST AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/Cognito_IdentityPoolAuth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......"
}
```

Respuesta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
  }
}
```

Qué ocurrió: Amazon Cognito validó el token con el proveedor configurado y emitió un token OpenID. La aplicación llamó AWS STS directamente para asumir una función de IAM y recibir credenciales temporales.

Funcionamiento del acceso social

- Los usuarios de redes sociales reciben AWS credenciales temporales a través de los grupos de identidades de Amazon Cognito después de autenticarse con su proveedor de redes sociales.
- Cada usuario autenticado recibe un ID de identidad único que permanece en todas las sesiones.
- Estas credenciales están vinculadas a un rol de IAM diseñado específicamente para el acceso autenticado, lo que proporciona permisos más amplios que el acceso como invitado.
- Los tokens de los proveedores de redes sociales se intercambian por AWS credenciales, lo que permite conservar la identidad y los permisos del usuario.

Autenticación con grupos de usuarios de Amazon Cognito

En este paso, se analiza la autenticación de Amazon Cognito con la integración del inicio de sesión administrado con grupos de usuarios. Al vincular un grupo de usuarios como un IdP a un grupo de identidades, los tokens del grupo de usuarios autorizan a su grupo de identidades a emitir credenciales temporales.

User pool authentication with enhanced flow

El flujo mejorado proporciona un enfoque simplificado para obtener credenciales de AWS a través de grupos de identidades de Amazon Cognito con una sola solicitud de API.

Utilice la autenticación del grupo de usuarios de Amazon Cognito con el flujo mejorado del grupo de identidades

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Elija la pestaña Acceso autenticado.

3. Seleccione la pestaña Flujo mejorado.
4. Elija Iniciar sesión con los grupos de usuarios de Amazon Cognito
5. Complete el inicio de sesión con su nombre de usuario y contraseña en el inicio de sesión administrado.
6. El grupo de usuarios redirige de nuevo al URI de redirección de su aplicación con un código de autorización.
7. La aplicación intercambia el código de autorización con su grupo de usuarios por tokens web JSON.
8. La aplicación intercambia el token de identificación con tu grupo de identidades por AWS credenciales temporales
9. La aplicación muestra el panel de resultados en la interfaz web

   Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

   a. Botón Ver solo credenciales: seleccione este botón si desea ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-12T13:36:17-07:00"
     }
   }
   ```

   b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

      - La aplicación inicia sesión en el usuario con un Amazon Cognito. Tras la autenticación correcta con el grupo de usuarios, la aplicación recibe una respuesta de OAuth 2.0 que contiene el token de identificación (JWT). Los grupos de identidades aceptan los tokens de ID de JWT de los grupos de usuarios que utilizan este formato de clave de proveedor:

Proveedor de identidades	Clave de proveedor de Cognito	Finalidad
Grupo de usuarios de Amazon Cognito	`cognito-idp.{region}.amazonaws.com/{user-pool-id}`	Tokens de ID de JWT de grupos de usuarios de Amazon Cognito

Tras la autenticación correcta con el grupo de usuarios, la aplicación recibe una respuesta de OAuth 2.0 que contiene el token de identificación (JWT):

```
{
   "id_token": "eyJraWQiOiJFWAMPLE......",
   "token_type": "Bearer",
   "expires_in": 3600
}
```

- Solicitud de API `GetId()` con su `identityPoolId` y un mapa `Logins` que incluye la clave del proveedor del grupo de usuarios asignada al `id_token`. Amazon Cognito verificó que la firma, el emisor, el vencimiento y la audiencia (`aud`) del token de ID del grupo de usuarios coincidan con uno de los ID de cliente de aplicación que usted ha registrado para este IdP del grupo de usuarios en el grupo de identidades.

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:1ac4a76d-1fef-48aa-83af-4224799c0b5c",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

Si es válido, busca o crea y devuelve el `IdentityID` del usuario. Un ejemplo de respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
```

```
}
```

- `GetCredentialsForIdentity()` con el `identityPoolId` devuelto y a=el mismo mapa `Logins` con el `id_token`. Amazon Cognito revalida que la firma, el emisor, el vencimiento y la audiencia (`aud`) del token de ID del grupo de usuarios coincidan con uno de los ID de cliente de aplicación que usted ha registrado para este IdP del grupo de usuarios en el grupo de identidades.

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

Si es válido, elige un rol de IAM (rol dentro del token, por reglas o predeterminado), llama a AWS STS en su nombre y devuelve credenciales de AWS temporales. Un ejemplo de respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "ASIAW7TIP7EJEXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
  }
}
```

User pool authentication with basic flow

El flujo básico proporciona un control pormenorizado del proceso de autenticación, con solicitudes de API independientes para la recuperación de la identidad y la generación de credenciales.

Utilice la autenticación del grupo de usuarios de Amazon Cognito con el flujo básico del grupo de identidades

1. En la aplicación de demostración, vaya a la sección Demostración interactiva

2. Elija la pestaña Acceso autenticado.
3. Seleccione la pestaña Flujo básico.
4. Elija Iniciar sesión con los grupos de usuarios de Amazon Cognito
5. Complete el inicio de sesión con su nombre de usuario y contraseña en el inicio de sesión administrado.
6. El grupo de usuarios redirige de nuevo al URI de redirección de su aplicación con un código de autorización.
7. La aplicación intercambia el código de autorización con su grupo de usuarios por tokens web JSON.
8. La aplicación intercambia el token de identificación con tu grupo de identidades por AWS credenciales temporales
9. La aplicación muestra el panel de resultados en la interfaz web

   Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

   a. Botón Ver solo credenciales: seleccione este botón si desea ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-12T13:36:17-07:00"
     }
   }
   ```

   b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

      - La aplicación inicia sesión en el usuario con un grupo de usuarios de Amazon Cognito y obtiene el token de identificación (JWT) como artefacto. Tras la autenticación correcta con el grupo de usuarios, la aplicación recibe una respuesta de OAuth que contiene el token de identificación (JWT). Los grupos de identidades utilizan este token para la autenticación:

      ```
      {
      ```

```
   "id_token": "eyJraWQiOiJFWAMPLE......",
   "token_type": "Bearer",
   "expires_in": 3600
}
```

- Solicitud de API `GetId()` con su ID de grupo de identidades y un mapa `Logins` que incluye su clave de proveedor de grupo de usuarios y el token de ID como valor. Amazon Cognito verificó que la firma, el vencimiento y la audiencia (aud) del token de ID del grupo de usuarios coincidan con uno de los ID de cliente de aplicación que usted ha registrado para este IdP del grupo de usuarios en el grupo de identidades.

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:1ac4a76d-1fef-48aa-83af-4224799c0b5c",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

  Si es válido, busca o crea y devuelve el `IdentityID` del usuario. Un ejemplo de respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetOpenIdToken()` con el `IdentityID` devuelto y el mismo mapa `Logins`

```
POST GetOpenIdToken
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "cognito-idp.us-east-1.amazonaws.com/us-east-1_EXAMPLE123":
 "eyJraWQiOiJFWAMPLE......"
  }
}
```

  Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

Qué ocurre en este paso: Amazon Cognito emite un token de identidad web OpenID Connect de corta duración desde cognito-identity.amazonaws.com que representa a este `IdentityId`. El token incluye las afirmaciones del OIDC que AWS STS evalúan, como aud (el ID de tu grupo de identidades) y amr (autenticado o no autenticado). La política de confianza de su rol de IAM debe exigir esas reclamaciones.

- `AssumeRoleWithWebIdentity()`- Tu aplicación llama AWS STS directamente para cambiar el token OpenID de Amazon Cognito por credenciales temporales AWS

```
POST sts:AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/Cognito_IdentityPoolAuth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......",
  "RoleSessionName": "CognitoIdentityCredentials"
}
```

Respuesta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "FwoGZXIvYXdzEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AROAW7TIP7EJYEXAMPLE:CognitoIdentityCredentials",
    "Arn": "arn:aws:sts::111122223333:assumed-role/
Cognito_IdentityPoolAuth_Role/CognitoIdentityCredentials"
  }
}
```

Qué hizo su aplicación de demostración: su aplicación envió el token OpenID desde `GetOpenIdToken()` hasta AWS STS, solicitando credenciales temporales. AWS STS realizó comprobaciones de validación y emitió credenciales:

Funcionamiento del acceso al grupo de usuarios

- Los usuarios del grupo de usuarios reciben AWS credenciales temporales a través de los grupos de identidades de Amazon Cognito.
- Estas credenciales están vinculadas a un rol de IAM especificado en la configuración del grupo de identidades.
- Los identificadores del grupo de usuarios se intercambian por AWS credenciales a través del grupo de identidades.

Autenticación SAML

En este paso, se analiza la autenticación SAML. Los usuarios pueden iniciar sesión con proveedores de identidad empresariales que admitan SAML para acceder a AWS los servicios. La aplicación de ejemplo no admite el flujo básico con SAML.

SAML authentication with enhanced flow

En esta sección, se muestra cómo puede usar un proveedor de identidades de SAML para iniciar sesión en un usuario y, mediante el flujo mejorado, intercambiar la afirmación de SAML en un grupo de identidades de Amazon Cognito por credenciales temporales AWS para llamar a los servicios. AWS

Uso de la autenticación SAML con el flujo mejorado del grupo de identidades

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Elija la pestaña Acceso autenticado.
3. Seleccione la pestaña Flujo mejorado.
4. Seleccione Iniciar sesión con el proveedor SAML
5. Complete el inicio de sesión con sus credenciales empresariales.
6. El grupo de usuarios redirige de nuevo al URI de redirección de su aplicación con una aserción SAML.
7. La aplicación intercambia el código de autorización con su grupo de usuarios por tokens web JSON.
8. La aplicación intercambia la respuesta de SAML con su grupo de identidades por credenciales temporales AWS
9. La aplicación muestra el panel de resultados en la interfaz web

Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

a. Botón Ver solo credenciales: seleccione este botón si desea ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

   - La aplicación inicia sesión en el usuario con un IdP SAML y obtiene la respuesta SAML. Los grupos de identidades aceptan las aserciones SAML de los proveedores empresariales que utilizan el ARN del proveedor SAML como clave:

Proveedor de identidades	Clave de proveedor de Cognito	Finalidad
Proveedor SAML	`arn:aws:iam::111122223333:saml-provider/EXAMPLE`	Afirmaciones de SAML de la empresa IdPs

   Tras la autenticación correcta con el proveedor de SAML, su aplicación recibe una respuesta de SAML mediante HTTP POST en la URL de devolución de llamada:

```
{
  "saml_response": "PD94bWwgdmVyc2lvbj0iMS4wIiBFWAMPLE...",
  "provider_arn": "arn:aws:iam::111122223333:saml-provider/EXAMPLE",
  "status": "Authentication successful"
}
```

- La solicitud de API `GetId()` con el ID de su grupo de identidades y un mapa `Logins` que contiene la aserción y el ARN del proveedor SAML

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "arn:aws:iam::111122223333:saml-provider/EXAMPLE":
 "PD94bWwgdmVyc2lvbj0iMS4wIiBFWAMPLE..."
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetCredentialsForIdentity()` con el `IdentityID` devuelto y el mismo mapa `Logins`

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "arn:aws:iam::111122223333:saml-provider/EXAMPLE":
 "PD94bWwgdmVyc2lvbj0iMS4wIiBFWAMPLE..."
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......"
  }
```

```
}
```

Qué ocurrió: Amazon Cognito validó la afirmación de SAML comparándola con el proveedor configurado, eligió un rol de IAM en función de los atributos o reglas de SAML y llamó en su nombre. AWS STS

Funcionamiento del acceso SAML

- Los usuarios empresariales reciben AWS credenciales temporales de los grupos de identidades de Amazon Cognito tras autenticarse con su proveedor de SAML.
- Cada usuario autenticado recibe un ID de identidad único que permanece en todas las sesiones.
- Estas credenciales están vinculadas a un rol de IAM diseñado específicamente para el acceso autenticado, lo que proporciona permisos más amplios que el acceso como invitado.
- Las afirmaciones de SAML se intercambian por AWS credenciales, manteniendo la identidad del usuario y los atributos empresariales.

Autenticación de OpenID Connect (OIDC)

En este paso, se analiza la autenticación OIDC con proveedores de identidad empresariales. Los usuarios pueden iniciar sesión a través del proveedor de identidad empresarial de su organización (como Azure AD, Okta o Google Workspace) para acceder a los servicios. AWS Si busca información sobre cómo integrar la autenticación basada en estándares con sus recursos de AWS , siga leyendo. Después de implementar la autenticación OIDC, aprenderá a aprovechar las aserciones OIDC para lograr un control de acceso detallado.

OIDC authentication with enhanced flow

En esta sección, se muestra cómo puede utilizar un proveedor de identidades OIDC para iniciar sesión en un usuario y, mediante el flujo mejorado, intercambiar el token OIDC de un grupo de identidades de Amazon Cognito por credenciales temporales para llamar a los servicios. AWS AWS

Uso del inicio de sesión de OIDC con el flujo mejorado del grupo de identidades

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Elija la pestaña Acceso autenticado.

3. Seleccione la pestaña Flujo mejorado.
4. Elija Iniciar sesión con el proveedor OIDC
5. Complete el inicio de sesión con sus credenciales empresariales.
6. El proveedor OIDC redirige de nuevo a la aplicación con un código de autorización
7. La aplicación intercambia el código de autorización con su grupo de usuarios por tokens web JSON.
8. La aplicación envía el token OIDC a su grupo de identidades y recupera las credenciales temporales. AWS
9. La aplicación muestra el panel de resultados en la interfaz web

   Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

   a. Botón Ver solo credenciales: seleccione este botón si desea ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

   ```
   {
     "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
     "Credentials": {
       "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
       "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
       "Expiration": "2025-08-12T13:36:17-07:00"
     }
   }
   ```

   b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

      - La aplicación inicia sesión en el usuario con un IdP de OIDC y obtiene el token de ID. Los grupos de identidades aceptan tokens OIDC de proveedores empresariales:

Proveedor de identidades	Clave de proveedor de Cognito	Finalidad
Proveedor OIDC	`example-provider.com/oauth2/default`	Tokens de ID OIDC de empresa IdPs

Tras la autenticación correcta con el proveedor OIDC, la aplicación recibe una respuesta de OAuth 2.0 que contiene los tokens:

```
{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "eyJraWQiOiJFWAMPLE......",
  "scope": "email openid profile",
  "id_token": "eyJraWQiOiJFWAMPLE......"
}
```

- La solicitud de API `GetId()` con el ID de su grupo de identidades y un mapa `Logins` que contiene el token de su proveedor OIDC

```
POST GetId
{
  "AccountId": "111122223333",
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetCredentialsForIdentity()` con el `IdentityID` devuelto y el mismo mapa Logins

```
POST GetCredentialsForIdentity
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......"
  }
}
```

Qué ocurrió: Amazon Cognito validó el token OIDC con el proveedor configurado, eligió un rol de IAM (predeterminado, basado en reclamos o mapeado por reglas) y llamó en su nombre. AWS STS

OIDC authentication with basic flow

En esta sección se muestra cómo puede utilizar un proveedor de identidades OIDC para iniciar sesión en un usuario y, mediante el flujo básico, intercambiar el token OIDC de un grupo de identidades de Amazon Cognito por credenciales temporales para llamar a los servicios. AWS AWS

Uso del inicio de sesión de OIDC con el flujo básico del grupo de identidades

1. En la aplicación de demostración, vaya a la sección Demostración interactiva
2. Elija la pestaña Acceso autenticado.
3. Seleccione la pestaña Flujo básico.
4. Elija Iniciar sesión con el proveedor OIDC
5. Complete el inicio de sesión con sus credenciales empresariales.
6. El proveedor OIDC redirige de nuevo a la aplicación con un código de autorización
7. La aplicación intercambia el código de autorización con su grupo de usuarios por tokens web JSON.
8. La aplicación envía el token OIDC a su grupo de identidades y recupera las credenciales temporales. AWS
9. La aplicación muestra el panel de resultados en la interfaz web

Cuando la autenticación se haya realizado correctamente, verá en la interfaz web el panel de resultados y tendrá dos opciones para explorarlos:

a. Botón Ver solo credenciales: seleccione este botón si desea ver directamente las AWS credenciales temporales generadas sin los detalles del flujo de la API.

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjEEXAMPLE......",
    "Expiration": "2025-08-12T13:36:17-07:00"
  }
}
```

b. El botón Ver flujo de API detallado: seleccione este botón si quiere ver las solicitudes de API detalladas.

   - La aplicación inicia sesión en el usuario con un IdP de OIDC y obtiene el token de ID. Los grupos de identidades aceptan tokens OIDC de proveedores empresariales:

Proveedor de identidades	Clave de proveedor de Cognito	Finalidad
Proveedor OIDC	`example-provider.com/oauth2/default`	Tokens de ID OIDC de empresa IdPs

   Tras la autenticación correcta con el proveedor OIDC, la aplicación recibe una respuesta de OAuth 2.0 que contiene los tokens:

```
{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "eyJraWQiOiJFWAMPLE......",
  "scope": "openid email profile",
  "id_token": "eyJraWQiOiJFWAMPLE......"
}
```

- La solicitud de API `GetId()` con el ID de su grupo de identidades y un mapa `Logins` que contiene el token de su proveedor OIDC

```
POST GetId
{
  "IdentityPoolId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

- `GetOpenIdToken()` con el IdentityID devuelto y el mismo mapa `Logins`

```
POST GetOpenIdToken
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Logins": {
    "example-provider.com/oauth2/default": "eyJraWQiOiJFWAMPLE......"
  }
}
```

Respuesta:

```
{
  "IdentityId": "us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Token": "eyJraWQiOiJFWAMPLE......"
}
```

- `AssumeRoleWithWebIdentity()` con el token OpenID

```
POST AssumeRoleWithWebIdentity
{
  "RoleArn": "arn:aws:iam::111122223333:role/Cognito_IdentityPoolAuth_Role",
  "WebIdentityToken": "eyJraWQiOiJFWAMPLE......"
```

```
}
```

Respuesta:

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken": "FwoGZXIvYXdzEEXAMPLE......",
    "Expiration": "2025-08-12T14:36:17-07:00"
  }
}
```

Qué ocurrió: Amazon Cognito validó el token OIDC con el proveedor configurado y devolvió un token OpenID. La aplicación llamó AWS STS directamente para asumir la función de IAM adecuada y recibió credenciales de corta duración.

Funcionamiento de la autenticación OIDC

- Standards-based: El OIDC se basa en OAuth 2.0 y proporciona información de identidad estandarizada.
- Validación de tokens: se puede validar la autenticidad de los tokens de ID.
- Claims-based acceso: el OIDC afirma que se puede utilizar para el mapeo de roles y el control de acceso.
- Integración empresarial: funciona con los proveedores de identidad empresarial más populares.

Siguientes pasos

Ahora que ha configurado y explorado la aplicación de demostración, puede hacer lo siguiente:

- Configurar proveedores de identidad adicionales que aún no haya probado
- Experimentar con la autenticación básica y la mejorada para comprender sus diferencias
- Personalizar la demostración para su propio caso de uso
- Integrar los grupos de identidades de Amazon Cognito en sus propias aplicaciones.

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. Third-party los auditores 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.

Para obtener información sobre las protecciones específicas del grupo de usuarios contra el registro, el inicio de sesión y la apropiación de cuentas malintencionados (MFA, autenticación adaptativa y protección contra amenazas), consulte. Protección de grupos de usuarios con protección contra amenazas

Contenido

- Protección de datos en Amazon Cognito
- Identity and access management para Amazon Cognito
- Registro y monitoreo en Amazon Cognito
- Acceda a Amazon Cognito mediante un punto final de interfaz (AWS PrivateLink)
- Validación de la conformidad para Amazon Cognito

- Resiliencia en Amazon Cognito
- Seguridad de la infraestructura en Amazon Cognito
- Configuración y análisis de vulnerabilidades en grupos de usuarios de Amazon Cognito
- AWS políticas gestionadas para Amazon Cognito

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. Eres 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:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos.
- 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 de los grupos de usuarios y grupos de identidades de Amazon Cognito se cifran en reposo de acuerdo con los estándares del sector.

- Amazon Cognito cifra los datos de los clientes en grupos de identidades con AWS claves propias. No puede cambiar este comportamiento.
- De forma predeterminada, Amazon Cognito también cifra los datos de los clientes en los grupos de usuarios con AWS claves propias. También puede configurar sus grupos de usuarios para, en su lugar, cifrar la información de sus clientes con claves administradas por el cliente.

AWS clave propia

Amazon Cognito cifra los datos de su grupo de usuarios o grupo de identidades con una clave de KMS propia AWS . Las claves de este tipo no están visibles en. AWS KMS

Clave administrada por el cliente

Amazon Cognito cifra los datos de su grupo de usuarios con una clave gestionada por el cliente. Usted es el propietario de la administración de las políticas clave gestionadas por el cliente, de la rotación y de la eliminación programada.

> **Note**
>
> Es posible que el cifrado con claves administradas por el cliente no esté disponible en algunos grupos de usuarios. Newly-created Los grupos de usuarios siempre disponen de esta forma de cifrado.

Lo que debe saber sobre el cifrado de grupos de usuarios con claves administradas por el cliente

1. Todos los datos de los clientes de los grupos de usuarios se cifran en reposo, incluso si no se toman medidas para configurar los ajustes de cifrado.
2. Amazon Cognito solo admite claves KMS simétricas en el mismo Región de AWS grupo de usuarios para el cifrado del grupo de usuarios en reposo. No puede configurar el cifrado de grupos

de usuarios en reposo con claves asimétricas. Puede configurar el cifrado en reposo con claves de una sola región y con claves de varias regiones que estén en la misma región que su grupo de usuarios.

3. Puede configurar el cifrado del grupo de usuarios solo con un ARN de clave KMS, no con un alias.

Cifrado de PII

Amazon Cognito respalda la confidencialidad, integridad y disponibilidad de la información de identificación personal (PII) en las búsquedas de atributos de los usuarios con un cifrado que permite realizar búsquedas. Estas funciones del código de autenticación de Hash-based mensajes (HMAC), optimizadas para el rendimiento de los conjuntos de datos de grupos de usuarios, mapean entre los valores cifrados y en texto plano de los atributos de los usuarios. Amazon Cognito calcula los valores de HMAC con la clave KMS que cifra el grupo de usuarios. Esta protección se aplica a los siguientes atributos:

- `sub`
- `email`
- `phone_number`
- `given_name`
- `family_name`
- `name`
- `username`
- `preferred_username`
- `cognito:user_status`

Los siguientes procedimientos configuran el cifrado en reposo en el grupo de usuarios. Para obtener más información sobre las políticas clave de KMS que delegan el acceso, Servicios de AWS como Amazon Cognito, consulte Permisos para Amazon Cognito en las políticas clave.

Set customer managed key policy

Para utilizar una clave gestionada por el cliente, la clave debe confiar en un director de servicio de Amazon Cognito para realizar las operaciones de cifrado y descifrado de la clave. Configure la política de claves de su clave KMS como se muestra en el siguiente ejemplo. El director de IAM que redacta esta política debe tener acceso de escritura a su clave de KMS, con `kms:PutKeyPolicy` permiso.

```
{
    "Id": "cognito-cmk-policy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Amazon Cognito service access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow Amazon Cognito service DescribeKey access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
```

```
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow administrators to manage the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

Set customer managed key policy with permissions for encryption of exported logs

Aplique esta política a su clave de KMS cuando haya configurado la exportación de registros. Esta política permite a Amazon Cognito cifrar los registros a medida que los exporta y al servicio intermedio descifrarlos antes de escribirlos en su CloudWatch grupo de registros. Configure la política de claves de su clave KMS como se muestra en el siguiente ejemplo. El director de IAM que redacta esta política debe tener acceso de escritura a su clave de KMS, con `kms:PutKeyPolicy` permiso.

```
{
    "Id": "cognito-cmk-policy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Amazon Cognito service access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow Amazon Cognito service DescribeKey access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
```

```
                    "cognito-idp.amazonaws.com",
                    "identitystore.amazonaws.com"
                ]
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": [
                        "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-
east-1_EXAMPLE"
                    ]
                },
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "Allow access through Amazon Cognito for all principals in
 account that are authorized to use Amazon Cognito",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKeyWithoutPlainText"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "cognito-idp.us-east-1.amazonaws.com"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cognito-idp:userpool-arn":
 "arn:aws:cognito-idp:us-east-1:111122223333:userpool/us-east-1_EXAMPLE"
                }
            }
        },
        {
            "Sid": "Allow Amazon Cognito service log delivery access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Encrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:SourceArn": "arn:aws:logs:us-
east-1:111122223333:*"
                }
            }
        },
        {
```

```
            "Sid": "Allow IngestionHub service log delivery access",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:SourceArn": "arn:aws:logs:us-
east-1:111122223333:*"
                }
            }
        },
        {
            "Sid": "Allow administrators to manage the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

Configure encryption at rest in the console

Para configurar el cifrado en reposo en un grupo de usuarios

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. Seleccione el menú Configuración y vaya a la pestaña de seguridad del grupo de usuarios. Localice Encryption at rest y seleccione Editar.

5. En Tipo de clave, selecciona la clave propia o la clave AWS gestionada por el cliente.

   a. Si seleccionó la AWS clave propia, no se requiere ninguna configuración adicional.

   b. Si seleccionó la clave administrada por el cliente, introduzca el ARN de una clave de KMS en el ARN de la clave administrada por el cliente. También puede elegir Crear una AWS KMS clave y abrir una nueva ventana en la AWS KMS consola para crear una nueva clave KMS.

6. Seleccione Save changes (Guardar cambios).

Configure encryption at rest with the API

Establezca la configuración de la clave en una solicitud de UpdateUserPoolAPI CreateUserPoolo en una solicitud. El siguiente ejemplo parcial del cuerpo de la solicitud establece un grupo de usuarios para usar la clave administrada por el cliente proporcionada. Para ver un ejemplo de solicitud completo, consulte Ejemplos.

```
"KeyConfiguration": {
   "KeyType": "CUSTOMER_MANAGED_KEY",
   "KmsKeyArn": "arn:aws:kms:us-east-1:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
},
```

El siguiente ejemplo parcial del cuerpo de la solicitud establece que un grupo de usuarios utilice AWS una clave propia.

```
"KeyConfiguration": {
   "KeyType": "AWS_OWNED_KEY"
},
```

Si la DescribeUserPoolrespuesta no incluye ningún `KeyConfiguration` parámetro, el grupo de usuarios está configurado para cifrar los datos en reposo con una clave AWS propia.

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.
- Paquetes de cifrado con perfecto secreto directo (PFS), como el DHE (Ephemeral) o el ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Los grupos de usuarios y grupos de identidades de Amazon Cognito tienen operaciones de IAM-authenticated API no autenticadas y autorizadas por token. Algunas son operaciones de tipo plano de control para operaciones administrativas, como la configuración de un dominio de grupo de usuarios, y otras son operaciones de tipo plano de datos para la autenticación. Para obtener más información, consulte Lista de operaciones de API agrupadas por modelo de autorización. Todas las clases de operaciones de la API de Amazon Cognito comparten un espacio de nombres (`cognito-idp`para grupos de usuarios, grupos de identidades) y puntos de enlace de `cognito-identity` servicio. AWS Los puntos de enlace del servicio requieren el cifrado en tránsito con una versión TLS mínima de 1.2.

Los grupos de usuarios de Amazon Cognito alojan el inicio de sesión gestionado y la clásica interfaz de usuario alojada en dominios web que se ofrecen desde distribuciones de Amazon propiedad del servicio. CloudFront Amazon Cognito administra la configuración del cifrado en tránsito en esas distribuciones. Para obtener más información sobre la configuración de cifrado del inicio de sesión administrado, consulte la versión TLS en el capítulo sobre el inicio de sesión administrado.

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
- Autenticación con identidades
- Administración del acceso con políticas
- Cómo funciona Amazon Cognito con IAM
- Identity-based ejemplos de políticas para Amazon Cognito
- Solución de problemas de identidad y acceso de Amazon Cognito
- Uso de roles vinculados a servicios para Amazon Cognito

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte Solución de problemas de identidad y acceso de Amazon Cognito).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte Cómo funciona Amazon Cognito con IAM).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte Identity-based ejemplos de políticas para Amazon Cognito).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte Cómo iniciar sesión en la Cuenta de AWS en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Cuenta de AWS usuario raíz

Al crear una Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz, que tiene acceso completo a todos los Servicios de AWS recursos. Se recomiendaencarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de 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 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 empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. 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 con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales en la Guía del usuario de IAM.

Un grupo de IAM especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un Rol de IAM es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol cambiando de un rol de usuario a uno de IAM (consola) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte Información general de políticas de JSON en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Identity-based políticas

Identity-based las políticas son documentos de política de permisos de JSON que se adjuntan a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Identity-based las políticas pueden ser políticas integradas (integradas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte Selección entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Resource-based políticas

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los ejemplos incluyen 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. Debe especificar una entidad principal en una política basada en recursos.

Resource-based las políticas son políticas en línea que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCP): definen los permisos máximos disponibles para los recursos de las cuentas. Para obtener más información, consulte Políticas de control de recursos (RCP) en la Guía del usuario de AWS Organizations .
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener 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 se AWS determina 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
Identity-based políticas	Sí
Resource-based políticas	No
Acciones de políticas	Sí
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
Service-linked roles	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.

Identity-based políticas de Amazon Cognito

Compatibilidad con las políticas basadas en identidad: sí

Identity-based las políticas son documentos de política de permisos de JSON que puedes adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. 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 la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente 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. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de la política de JSON de IAM en la Guía del usuario de IAM.

Identity-based ejemplos de políticas para Amazon Cognito

Para ver ejemplos de políticas basadas en identidad de Amazon Cognito, consulte Identity-based ejemplos de políticas para Amazon Cognito.

Resource-based políticas de Amazon Cognito

Admite políticas basadas en recursos: no

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los 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 basada en 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. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Acciones de política de Amazon Cognito

Compatibilidad con las acciones 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 `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. 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 Lista de operaciones de API agrupadas por modelo de autorización.

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 Identity-based ejemplos de políticas para Amazon Cognito.

Recursos de políticas de Amazon Cognito

Compatibilidad con los 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. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). En el caso de las acciones que no admiten permisos por recurso, 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 Identity-based ejemplos de políticas para Amazon Cognito.

Claves de condición de políticas de Amazon Cognito

Compatibilidad con 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` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan operadores de condición, tales como

igual o menor que, para que la condición de la política coincida con los valores de la solicitud. 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 Identity-based ejemplos de políticas para Amazon Cognito.

Listas de control de acceso (ACL) en Amazon Cognito

Compatibilidad con 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.

Attribute-based control de acceso (ABAC) con Amazon Cognito

Compatibilidad con ABAC (etiquetas en las políticas): parcial

Attribute-based el control de acceso (ABAC) es una estrategia de autorización que define los permisos en función de unos atributos denominados etiquetas. Puede adjuntar etiquetas a las entidades y AWS los recursos de IAM y, a continuación, diseñar políticas de ABAC para permitir las operaciones cuando la etiqueta del director coincida con la etiqueta del recurso.

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 Definición de permisos con la autorización de 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

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a AWS los recursos a corto plazo y se crean automáticamente cuando se utiliza la federación o se cambia de rol. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM y Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Cross-service permisos principales para Amazon Cognito

Compatibilidad con sesiones de acceso directo (FAS): no

Las sesiones de acceso directo (FAS) utilizan los permisos del operador principal que llama y los que solicitan Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Sesiones de acceso directo.

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 Crear 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 Implementación de la 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.

Service-linked funciones 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 la función de realizar una acción en su nombre. Service-linked las funciones aparecen en su nombre 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.

Identity-based ejemplos de políticas para Amazon Cognito

De forma predeterminada, los usuarios y los roles no tienen permiso para crear ni modificar los recursos de Amazon Cognito. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan.

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 (consola) 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
- Uso de la consola de Amazon Cognito
- Cómo permitir a los usuarios consultar sus propios permisos
- Restricción del acceso a la consola a un grupo de identidades concreto
- Autorización del acceso a un conjunto de datos concreto para todas las identidades de un grupo

Prácticas recomendadas sobre las políticas

Identity-based las políticas determinan si alguien puede crear recursos de Amazon Cognito de su cuenta, acceder a ellos o eliminarlos. 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 tarea 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 CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza 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 Validación de políticas con el Analizador de acceso de IAM en la Guía del 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 exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte 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, permita el 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 obtener 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

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:ListIdentityPools"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:*"
```

```
            ],
            "Resource": "arn:aws:cognito-identity:us-
east-1:111122223333:identitypool/us-east-1:1a1a1a1a-ffff-1111-9999-12345678"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cognito-sync:*"
            ],
            "Resource": "arn:aws:cognito-sync:us-
east-1:111122223333: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

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-sync:ListRecords",
                "cognito-sync:UpdateRecords"
            ],
            "Resource": "arn:aws:cognito-sync:us-
east-1:111122223333: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 para acceder 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 la función 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 AWS administrador. 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 otras personas accedan a Amazon Cognito, debe conceder permiso a las personas o aplicaciones que lo necesiten. Si usa AWS IAM Identity Center para administrar las personas y las aplicaciones, debe asignar conjuntos de permisos a los usuarios o grupos para definir su nivel de acceso. Los conjuntos de permisos crean políticas de IAM y las asignan a los roles de IAM asociados a la persona o aplicación de forma automática. Para obtener más información, consulte la sección Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .

Si no utiliza IAM Identity Center, debe crear entidades de IAM (usuarios o roles) para las personas o aplicaciones que necesitan acceso. A continuación, debe asociar una política a la entidad que les adjudica los permisos correctos en Amazon Cognito. Una vez concedidos los permisos, proporcione las credenciales al usuario o al desarrollador de la aplicación. Utilizarán esas credenciales para acceder a AWS. Para obtener más información sobre la creación de usuarios, grupos, políticas y permisos de IAM, consulte Identidades de IAM y Políticas y permisos en IAM en la Guía del usuario de IAM.

Quiero permitir que personas ajenas a mi AWS cuenta para acceder a mis recursos de Amazon Cognito

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se 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 obtener 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 los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario 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 conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM 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 Service-linked Amazon Cognito predefine las funciones e incluyen todos los permisos que el servicio requiere para llamar a otros AWS 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 son compatibles con roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios en los que se

indica Sí en la columna Service-Linked Rol vinculado a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Service-linked permisos de rol 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 los permisos de los Service-linked roles 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**
>
> Al integrar Amazon Cognito con Amazon Pinpoint `createUserPoolClient` mediante `updateUserPoolClient` y, los permisos de recursos se añaden a la SLR como una política integrada. La política en línea proporciona permisos y permisos. `mobiletargeting:UpdateEndpoint` `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 Consola de administración de AWS 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 Consola de administración de AWS 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 Consola de administración de AWS 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).

Para eliminar manualmente el 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

- Supervisión y administración de costos
- Exportación de registros de grupos de usuarios de Amazon Cognito
- Seguimiento de las cuotas CloudWatch y el uso en Service Quotas
- Inicio de sesión en Amazon Cognito AWS CloudTrail

Supervisión y administración de costos

Al igual que con cualquier otro Servicio de AWS, es importante entender el efecto de la configuración y el uso de Amazon Cognito en la factura AWS . En los preparativos para la implementación de grupos de usuarios en la producción, configure la supervisión y las medidas de seguridad para controlar la actividad y el consumo de recursos. Cuando sepa dónde mirar y cuáles son las medidas que generan costos adicionales, podrá establecer medidas preventivas para evitar sorpresas en la factura.

Amazon Cognito cobra por los siguientes usos.

- Usuarios activos mensuales del grupo de usuarios (MAU): la tarifa varía según el plan de características
- MAU del grupo de usuarios que han iniciado sesión con la federación OIDC o SAML
- Solicita un volumen de autorizaciones de máquina a máquina (M2M), tanto desde el punto de `client_credentials` enlace del token como desde la operación de la API `GetClientToken`
- Adquisición de uso por encima de las cuotas predeterminadas para algunas categorías de API de grupos de usuarios

Además, las características del grupo de usuarios, como los mensajes de correo electrónico, los mensajes SMS y los desencadenadores de Lambda, pueden generar costos en servicios dependientes. Para obtener una descripción completa, consulte Precios de Amazon Cognito.

Visualización y previsión de los costos

High-volume eventos como el lanzamiento de productos y la apertura a nuevas bases de usuarios pueden aumentar el número de MAU y repercutir en los costes. Calcule el número de usuarios nuevos con antelación y observe la actividad a medida que se produce. Puede que decida ajustar el volumen comprando capacidad de cuota adicional o controlando el volumen con medidas de seguridad adicionales.

Puede ver sus AWS costes e informar sobre ellos en la Administración de facturación y costos de AWS consola. Encontrará su cargos más recientes de Amazon Cognito en la sección Facturación y pagos. En Facturas, Cargos por servicio, filtre en `Cognito` para ver el 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 de Service Quotas. Por ejemplo, tanto las solicitudes de `client_credentials` concesión al punto final del token como las solicitudes de GetClientTokenAPI se muestran como Tasa de ClientAuthentication solicitudes, ya que comparten una cuota de solicitudes. En la factura, estas solicitudes se asocian al cliente de aplicación que las ha generado. Con esta información, puede asignar los costos de manera equitativa a los inquilinos en una arquitectura de múltiples inquilinos.

Para hacer un recuento de las solicitudes M2M realizadas en puntos terminales durante un período de tiempo, también puede enviar AWS CloudTrail los eventos a Logs para su análisis. CloudWatch 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. Para incluir las solicitudes M2M que realice con la operación de la `GetClientToken` API, utilice la métrica de tasa de utilización de ClientAuthentication solicitudes descrita anteriormente en esta sección.

```
filter eventName = "Token_POST" and @message like '"grant_type":["client_credentials"]'
 | stats count(*)
```

Administración de costos

Amazon Cognito factura en función del número de usuarios, el uso de las características y el volumen de solicitudes. A continuación le proporcionamos algunos consejos para administrar los costos 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 la factura. Evite las operaciones que puedan activar a los usuarios. En lugar de la operación de AdminGetUserAPI, consulta a los usuarios con la ListUsersoperación. No realice con usuarios inactivos pruebas administrativas de gran volumen de operaciones de grupos de usuarios.

Vinculación de los usuarios federados

Los usuarios que inician sesión con un proveedor de identidades SAML 2.0 u OpenID Connect (OIDC) tienen un costo más elevado 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 se incluyen en 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.

Administración de las tarifas de solicitudes

Si su grupo de usuarios está rozando el límite superior de la cuota, sopese la posibilidad de adquirir capacidad adicional para administrar el volumen. También es posible que pueda reducir el volumen de solicitudes de la aplicación. Para obtener más información, consulte Optimización de las tasas de solicitudes para cumplir los límites de cuota.

Solicitud de un nuevo token solo cuando se necesite

La autorización de máquina a máquina (M2M) con concesiones de credenciales de cliente puede alcanzar un gran volumen de solicitudes de token. Cada nueva solicitud de token repercute en la cuota de solicitudes y en el importe de la factura. Para optimizar los costos, incluya la configuración de caducidad de los tokens y la gestión de tokens en el diseño de las 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. Al implementar este método, el proxy de almacenamiento en caché actúa como protección contra las aplicaciones que solicitan tokens de acceso sin saber que han caducado los tokens previamente adquiridos. El almacenamiento en caché de los tokens es ideal para microservicios de corta duración, como las funciones de Lambda o los contenedores de Docker.
- Implemente mecanismos de administración de tokens en las aplicaciones que tengan en cuenta la caducidad de los tokens. No solicite un nuevo token hasta que los anteriores estén a punto de caducar. Como práctica recomendada, actualice los tokens cuando haya transcurrido aproximadamente el 75 % de su vida útil. Esta práctica maximiza la duración del token y, al mismo tiempo, garantiza la continuidad del usuario en la aplicación.

  Evalúe las necesidades de confidencialidad y disponibilidad de cada aplicación y configure el cliente de aplicación del grupo de usuarios para que emita tokens de acceso con un período de validez adecuado. La duración del token personalizado funciona mejor con las API y los servidores de larga duración que pueden administrar de forma continua la frecuencia de las solicitudes de credenciales.

ListUsers, no AdminGetUser

Para consultar los atributos de los usuarios de su grupo de usuarios, utilice la operación de la ListUsersAPI y los métodos del SDK asociados siempre que sea posible. AdminGetUsermarca a un usuario como activo durante el mes y contribuye a los usuarios activos mensuales (MAU) que se utilizan para calcular la factura de los grupos de usuarios.

Administración de planes de características

Si elige un plan de características en un grupo de usuarios, la tarifa de facturación se aplicará a todos los MAU del grupo de usuarios. Si tiene usuarios que no necesitan características incluidas en un plan de características de nivel superior, sepárelos en otro grupo de usuarios.

Exportación de registros de grupos de usuarios de Amazon Cognito

Puede configurar su grupo de usuarios para enviar registros detallados de alguna actividad adicional a otro Servicio de AWS, como un grupo de CloudWatch registros. Estos registros son más detallados que los de su grupo y pueden resultar útiles para solucionar problemas de su grupo de usuarios y analizar la actividad de inicio de sesión de los usuarios con protección frente a amenazas. AWS CloudTrail Si desea transmitir registros de errores en las notificaciones por SMS y correo electrónico, su grupo de usuarios envía los registros de `ERROR` nivel a un grupo de registros. CloudWatch Cuando desee transmitir registros de la actividad de inicio de sesión de los usuarios, el grupo de usuarios envía los registros de nivel `INFO` a un grupo de registro, a un flujo de Amazon Data Firehose o a un bucket de Amazon S3. Puede combinar ambas opciones en un grupo de usuarios.

Temas

- Cosas que debe saber sobre la exportación de archivos de registro
- Errores de exportación de correos electrónicos y mensajes SMS
- Exportación de los registros de actividad de los usuarios en la protección contra amenazas

Cosas que debe saber sobre la exportación de archivos de registro

Impacto del costo

> Amazon Data Firehose, Amazon S3 y CloudWatch Logs conllevan costes por la ingesta y recuperación de datos. La configuración de registro puede afectar a su factura. AWS Para obtener más información, consulte los siguientes temas:

- CloudWatch Precios de registros vendidos en Amazon.
- Precios de Amazon Data Firehose
- Precios de Amazon S3

User-activity Las exportaciones de registros contienen evaluaciones de seguridad y son una función de la protección contra las amenazas del grupo de usuarios. Amazon Cognito solo genera estos registros cuando la protección contra amenazas está activa Audit-onlyo en Full-functionmodo y su grupo de usuarios está en el plan de características Plus.

Los registros de actividad de los usuarios están en el nivel **`INFO`**

Los registros de actividad de los usuarios exportados solo están en el nivel de error `INFO`, y proporcionan información para el análisis estadístico y de seguridad de la actividad de autenticación. Los mensajes en los niveles de error `WARNING` y `ERROR`, como los errores de limitación, no están incluidos en los registros exportados.

Best-effort entrega

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 Logs, Amazon S3 y Firehose pueden afectar a la entrega de registros.

Los registros externos existentes no se ven afectados

Estas opciones de registro no reemplazan ni cambian 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 puedes buscar los registros de Ver los resultados de importación del grupo de usuarios en la CloudWatch consola y dentro de Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda los CloudWatch registros. Amazon Cognito y Lambda almacenan estos registros en grupos de registros diferentes de los que especifica para obtener registros de actividad de usuarios.

Solo se aplica a los grupos de usuarios

No existen capacidades de exportación de registros para los grupos de identidades.

Se requieren permisos de usuario y un rol vinculado a servicios

El AWS director que configura la exportación de registros debe tener permisos para modificar los recursos de destino, tal y como se describe en los temas siguientes. Amazon Cognito crea un rol vinculado a servicios en su lugar y asume el rol de entregar los registros al recurso de destino.

Para obtener más información sobre el modelo de autorización para enviar registros desde Amazon Cognito, consulte Habilitar el registro desde Servicios de AWS en la Guía del usuario de Amazon CloudWatch Logs.

El nivel de registro es exclusivo del tipo de registro

Message-delivery los registros son del `userNotification` tipo y del nivel de `ERROR` error. Los registros de actividad de los usuarios de seguridad avanzada son del tipo `userAuthEvents` y del nivel de error `INFO`. Puede combinar dos miembros de`LogConfigurations`, uno para `userNotification` To CloudWatch Logs y otro `userAuthEvents` para Firehose, Amazon S3 o CloudWatch Logs.

No puede enviar los registros de actividad de los usuarios a varios destinos. No puedes enviar los registros de notificaciones a los usuarios a ningún otro destino que CloudWatch no sea Logs.

Opciones de configuración diferentes

Solo puede configurar registros de notificación de usuario con la API de grupos de usuarios de Amazon Cognito o un AWS SDK. Puede configurar registros de actividad de usuario de seguridad avanzada con la API o en la consola de Amazon Cognito. Para configurar ambos, usa la API como se muestra en la solicitud de ejemplo en SetLogDeliveryConfiguration.

Se requiere configuración adicional con políticas de recursos de gran tamaño

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.

Creación automática de una carpeta en Amazon S3

Al configurar la exportación del registro de protección contra amenazas en un bucket de Amazon S3, Amazon Cognito puede crear una carpeta `AWSLogs` en el bucket. Esa carpeta no se crea en todos los casos y la configuración se puede realizar correctamente sin crearla.

Errores de exportación de correos electrónicos y mensajes SMS

En el caso de que se produzcan errores de entrega de mensajes de correo electrónico y SMS, puede entregar registros de notificación de usuario de nivel Error del grupo de usuarios. Al activar esta característica, puede elegir el grupo de registros al que desea que Amazon Cognito envíe los registros. El registro de notificaciones de los usuarios es útil cuando desea conocer el estado de los mensajes de correo electrónico y SMS que el grupo de usuarios ha entregado con Amazon SNS y Amazon SES. Esta opción de exportación de registros, a diferencia de la exportación de la actividad del usuario, no requiere el plan de características Plus.

Puede configurar registros de notificaciones 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. A continuación, se muestra un ejemplo de cuerpo de la solicitud .

```
{
   "LogConfigurations": [
      {
         "CloudWatchLogsConfiguration": {
            "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-group:example-user-
pool-exported"
         },
         "EventSource": "userNotification",
         "LogLevel": "ERROR"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

Debe autorizar estas solicitudes con AWS credenciales que tengan los siguientes permisos.

JSON

```
{
    "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"
    }
}
```

Exportación de los registros de actividad de los usuarios en la protección contra amenazas

Los grupos de usuarios con el plan de características Plus de seguridad avanzada registran los eventos de actividad de los usuarios: los detalles y la evaluación de seguridad del inicio y el cierre de sesión de los usuarios y otras operaciones de autenticación con el grupo de usuarios. Es posible que desee revisar los registros de actividad de los usuarios en su propio sistema de administración de registros o crear un archivo. Puede exportar estos datos a un grupo de CloudWatch registros de Amazon Logs, a una transmisión de Amazon Data Firehose o a un bucket de Amazon Simple Storage Service (Amazon S3). A partir de ahí, puede incorporar estos datos a otros sistemas que analicen, normalicen o procesen los datos de forma que se adapten a los procesos operativos. Para exportar datos de este tipo, el grupo de usuarios debe estar en el plan de características Plus y la protección contra amenazas debe estar activa en el grupo de usuarios.

Con la información de estos registros de actividad de los usuarios, puede ver un perfil de las actividades de inicio de sesión y administración de cuentas de los usuarios. De forma predeterminada, Amazon Cognito captura estos eventos en el almacenamiento que se encuentra en su grupo de usuarios. El siguiente ejemplo muestra un evento de un usuario que ha iniciado sesión y se ha evaluado que no tenía factores de riesgo. Puede recuperar esta información con la operación de la API `AdminListUserAuthEvents`. El siguiente es un ejemplo de output:

```
{
    "AuthEvents": [
        {
            "EventId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "EventType": "SignIn",
            "CreationDate": "2024-06-27T10:49:59.139000-07:00",
            "EventResponse": "Pass",
            "EventRisk": {
                "RiskDecision": "NoRisk",
                "CompromisedCredentialsDetected": false
```

```
            },
            "ChallengeResponses": [
                {
                    "ChallengeName": "Password",
                    "ChallengeResponse": "Success"
                }
            ],
            "EventContextData": {
                "IpAddress": "192.0.2.1",
                "DeviceName": "Chrome 126, Windows 10",
                "Timezone": "-07:00",
                "City": "null",
                "Country": "United States"
            }
        }
    ],
    "NextToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222#2024-06-27T17:49:59.139Z"
}
```


Puede activar la exportación de registros para la actividad de los usuarios en la consola de Amazon Cognito o mediante la operación de SetLogDeliveryConfigurationAPI.

Consola de administración de AWS

1. Si aún no tienes uno que quieras usar, crea un bucket de S3, una transmisión Firehose o un grupo de CloudWatchregistros.
2. Inicie sesión en la consola de Amazon Cognito.
3. Elija User Pools (Grupos de usuarios).
4. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
5. Seleccione la pestaña Seguridad avanzada. Busque Exportar registros de actividad de los usuarios y seleccione Editar.
6. En Estado de registro, seleccione la casilla de verificación situada junto a Activar la exportación del registro de actividad del usuario.
7. En Destino del registro, elige el Servicio de AWS que quieres que gestione tus registros: grupo de CloudWatch registros, transmisión de Amazon Data Firehose o bucket de S3.
8. Su selección rellena el selector de recursos con el tipo de recurso correspondiente. Seleccione un grupo de registro, un flujo o un bucket de la lista. También puede seleccionar el botón Crear para ir al Consola de administración de AWS servicio seleccionado y crear un recurso nuevo.

9. Seleccione Guardar cambios.

API

Elija un tipo de destino para los registros de actividad de los usuarios.

A continuación se muestra un ejemplo de cuerpo de solicitud `SetLogDeliveryConfiguration` que establece un flujo de Firehose como destino del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "FirehoseConfiguration": {
            "StreamArn": "arn:aws:firehose:us-west-2:123456789012:deliverystream/
example-user-pool-activity-exported"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

A continuación se muestra un ejemplo de cuerpo de solicitud `SetLogDeliveryConfiguration` que establece un bucket de Amazon S3 como destino del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "S3Configuration": {
            "BucketArn": "arn:aws:s3:::amzn-s3-demo-logging-bucket"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

A continuación se muestra un ejemplo del cuerpo de una `SetLogDeliveryConfiguration` solicitud que establece un CloudWatch grupo de registros como destino del registro.

```
{
   "LogConfigurations": [
      {
         "EventSource": "userAuthEvents",
         "CloudWatchLogsConfiguration": {
            "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-group:DOC-
EXAMPLE-LOG-GROUP"
         },
         "LogLevel": "INFO"
      }
   ],
   "UserPoolId": "us-west-2_EXAMPLE"
}
```

El usuario que configura la entrega de registros debe ser administrador de un grupo de usuarios y tener los siguientes permisos adicionales:

Amazon S3

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageUserPoolLogs",
            "Action": [
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "ManageLogsS3",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "s3:PutBucketPolicy",
```

```
                "s3:GetBucketPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatch Logs

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageUserPoolLogs",
            "Action": [
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "ManageLogsCWL",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
```

```
    ]
}
```

Amazon Data Firehose

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageUserPoolLogs",
            "Action": [
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "ManageUserPoolLogsFirehose",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "iam:CreateServiceLinkedRole",
                "firehose:TagDeliveryStream"
            ],
            "Resource": "*"
        }
    ]
}
```

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_ACTIVITY",
```

```
    "logLevel": "INFO",
    "message": {
        "version": "1",
        "eventId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventType": "SignUp",
        "userSub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "userName": "test-user",
        "userPoolId": "us-west-2_EXAMPLE",
        "clientId": "1example23456789",
        "creationDate": "Wed Jul 17 17:25:55 UTC 2024",
        "eventResponse": "InProgress",
        "riskLevel": "",
        "riskDecision": "PASS",
        "challenges": [],
        "deviceName": "Other, Other",
        "ipAddress": "192.0.2.1",
        "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "idpName": "",
        "compromisedCredentialDetected": "false",
        "city": "Seattle",
        "country": "United States",
        "eventFeedbackValue": "",
        "eventFeedbackDate": "",
        "eventFeedbackProvider": "",
        "hasContextData": "true"
    },
    "logSourceId": {
        "userPoolId": "us-west-2_EXAMPLE"
    }
}
```


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, consulte Observabilidad entre cuentas de CloudWatch .

Temas

- Las métricas del grupo de usuarios se incluyen CloudWatch
- Métricas en Service Quotas

Las métricas del grupo de usuarios se incluyen CloudWatch

Los grupos de usuarios utilizan las estadísticas de actividad de los usuarios como métricas. CloudWatch Desde CloudWatch, puede analizar el volumen de la actividad de autenticación y el uso de las cuotas en sus grupos de usuarios. Con la información de estas métricas, puede configurar alarmas para los eventos importantes y ajustar la configuración del grupo de usuarios según sea necesario. Mientras que el registro de la actividad de los usuarios contiene registros detallados de la actividad de los usuarios en sus grupos de usuarios, CloudWatch las métricas tienen estadísticas e indicadores de rendimiento agregados.

En la siguiente tabla, se enumeran las métricas disponibles para grupos de usuarios de Amazon Cognito. Amazon Cognito publica métricas en los espacios de nombres `AWS/Cognito` y `AWS/Usage`. Para obtener más información, consulta la Guía del CloudWatch usuario de Namespaces in Amazon.

Para obtener más información sobre el seguimiento de cuotas y uso, consulte Seguimiento del uso de cuotas y Seguimiento de los usuarios activos mensuales (MAU).

> **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	Description (Descripción)	Namespace
`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 limitada también se considera una solicitud fallida y, por lo tanto, una solicitud limitada también produce un recuento de 0. Para encontrar el porcentaje de solicitudes de registro de usuarios correctas, utilice	`AWS/Cognito`

Métrica	Description (Descripción)	Namespace
	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 solicitud es de registro de usuarios fallidas, utilice la CloudWatc h `Math` expresión y reste la estadística de la `Sum` estadísti ca. `Sample Count` 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`, `UserPoolC lient`	

Métrica	Description (Descripción)	Namespace
	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	`AWS/Cognito`

Métrica	Description (Descripción)	Namespace
SignInSuccesses	Proporciona la cantidad total de solicitudes de autentica ción de usuarios correctas realizadas al grupo de usuarios de Amazon Cognito. Una autenticación de usuario se considera correcta cuando se emite un token de autentica ción al usuario. Una autentica ción correcta produce un valor de 1, mientras que una solicitud incorrecta produce un valor de 0. Una solicitud limitada también se considera una solicitud fallida y, por lo tanto, una solicitud limitada también produce 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 autentica ción de usuario correctas, utilice la estadística Sum en esta métrica. Para contar el número total de solicitudes de autenticación de usuario	AWS/Cognito

Métrica	Description (Descripción)	Namespace
	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. 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	Description (Descripción)	Namespace
`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	`AWS/Cognito`

Métrica	Description (Descripción)	Namespace
	estadísticas de token de `Refresh`. Dimensión de métrica: `UserPool`, `UserPoolClient` Unidades: recuento	

Métrica	Description (Descripción)	Namespace
`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 limitada también se considera una solicitud fallida y, por lo tanto, una solicitud limitada también produce 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 estadística `Sum` en esta métrica. Para contar el número total de solicitudes	`AWS/Cognito`

Métrica	Description (Descripción)	Namespace
	fallidas para actualizar un token de Amazon Cognito, utilice la CloudWatch `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`, `UserPoolClient` Unidades: recuento	

Métrica	Description (Descripción)	Namespace
`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.	`AWS/Cognito`

Métrica	Description (Descripción)	Namespace
	Dimensión de métrica: `UserPool`, `UserPoolClient` Unidades: recuento	

Métrica	Description (Descripción)	Namespace
`FederationSuccesses`	Proporciona la cantidad total de solicitudes de identidad federada correctas al grupo de usuarios de Amazon Cognito. Se considera que una federación de identidad es se ha realizado correctam ente cuando Amazon Cognito emite tokens de autentica ción 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 estadística `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 CloudWatc	`AWS/Cognito`

Métrica	Description (Descripción)	Namespace
	h `Math` expresión y reste la estadística de la `Sum` estadísti ca. `Sample Count` Dimensión de métrica: `UserPool`, `UserPoolC lient` , `IdentityP rovider` Unidades: recuento	
`FederationThrottles`	Proporciona la cantidad total de solicitudes de identidad federada con limitación controlada 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 estadísti ca `Sum` para esta métrica. Dimensión de métrica: `UserPool`, `UserPoolC lient` , `IdentityP rovider` Unidades: recuento	`AWS/Cognito`

Métrica	Description (Descripción)	Namespace
`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 limitación controlada, llamadas fallidas y llamadas correctas. 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	`AWS/Usage`

Métrica	Description (Descripción)	Namespace
`ThrottleCount`	Proporciona la cantidad total de llamadas con limitación controlada en relación con una categoría. 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	`AWS/Usage`

Visualización de las métricas de protección contra amenazas

Las métricas que publica el grupo de usuarios contienen información estadística sobre el efecto que la configuración de protección contra amenazas aplicada tiene sobre la actividad de autenticación de los usuarios. Puede que quiera saber cuántos usuarios intentan iniciar sesión con credenciales comprometidas. También puede averiguar qué porcentaje de la actividad de inicio de sesión se ha considerado que presenta algún nivel de riesgo. Amazon Cognito publica métricas sobre las funciones de protección contra amenazas en su cuenta de Amazon. CloudWatch Amazon Cognito agrupa las métricas de protección contra amenazas por nivel de riesgo y también por nivel de solicitud.

Para añadir contexto al análisis de riesgos, puede ver información sobre los intentos de inicio de sesión de usuarios individuales, ya sea en su grupo de usuarios o en un origen de datos exportado.

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. Se puede hacer lo siguiente:

   - Para ordenar la tabla, utilice el encabezado de columna.
   - Para representar gráficamente una métrica, seleccione la casilla situada junto a la métrica. Para seleccionar todas las métricas, seleccione la casilla de verificación situada en la fila de encabezados 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	Description (Descripción)	Dimensiones de la métrica	Namespace
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 (predeterminado), medio o bajo.	`AWS/Cognito`
AccountTakeoverRisk	Solicitudes en las que Amazon Cognito detectó riesgo de	Operation: tipo de operación. `PasswordChange`, `SignIn` o `SignUp`	`AWS/Cognito`

Métrica	Description (Descripción)	Dimensiones de la métrica	Namespace
	usurpación de la cuenta.	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.	`AWS/Cognito`
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.	`AWS/Cognito`

Métrica	Description (Descripción)	Dimensiones de la métrica	Namespace
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.	`AWS/Cognito`

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	Description (Descripción)
By Risk Classification	Solicitudes que Amazon Cognito identifica como arriesgadas.
By Request Classification	Métricas agregadas por solicitud.

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`.

Dimensión	Descripción
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.
Clase	La clase de recurso a la que se realiza el seguimiento. Amazon Cognito no utiliza la dimensión de clase.

Utilice 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 muestra las métricas de todos los AWS servicios que utiliza. 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. Selecciona la casilla de verificación situada junto a una métrica. Las métricas aparecen 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 .

Crea 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.

Métricas en Service Quotas

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, que te llevará a las métricas relacionadas en la CloudWatch consola.

Inicio de sesión en Amazon Cognito 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 configurar CloudWatch para que 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 que Amazon Cognito envía a CloudTrail
- Análisis de CloudTrail eventos de Amazon Cognito con Amazon Logs Insights CloudWatch
- Ejemplo de evento de Amazon Cognito

Información que Amazon Cognito envía a 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 lo siguiente:

- 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 userIdentity de CloudTrail .

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_DUE_TO_SECURITY_REASONS` Para ver ejemplos de campos que Amazon Cognito no rellena para los eventos, consulte Ejemplo de evento 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 sus solicitudes de API.

Eventos de grupos de usuarios

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 ha realizado 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 páginas de inicio de sesión gestionadas. 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 en sus servicios de inicio de sesión gestionado como eventos en. CloudTrail

Hosted UI (classic) events

Eventos de interfaz de usuario (clásicos) alojados en CloudTrail

Operación	Description (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`.

Operación	Description (Descripción)
`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.
`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.

Operación	Description (Descripción)
`UserInfo_GET` , `UserInfo_POST`	Un usuario o proveedor de identidad intercambia información con su El punto de conexión userInfo.
`Confirm_With_Link_GET`	Un usuario envía una confirmación basada en un enlace que Amazon Cognito envió en un mensaje de correo electrónico.
`Event_Feedback_GET`	Un usuario envía comentarios a Amazon Cognito sobre un evento de protección contra amenazas.

Managed login events

Eventos de inicio de sesión gestionados en CloudTrail

Operación	Description (Descripción)
`login_POST`	Un usuario envía credenciales a su Punto de conexión Login.
`login_continue_POST`	Un usuario que ya ha iniciado sesión una vez elige volver a iniciar sesión.
`forgotPassword_POST`	Un usuario restablece su contraseña.
`selectChallenge_POST`	Un usuario responde a un desafío de autenticación después de enviar su nombre de usuario o sus credenciales.
`confirmUser_GET`	Un usuario abre el enlace en un mensaje de correo electrónico de confirmación o verificación.
`mfa_back_POST`	Un usuario selecciona el botón Volver después de una petición de MFA.

Operación	Description (Descripción)
`mfa_options_POST`	Un usuario selecciona una opción MFA.
`mfa_phone_register_POST`	Un usuario envía un número de teléfono para registrarlo como factor de MFA. Esta operación hace que Amazon Cognito envíe un código MFA a su número de teléfono.
`mfa_phone_verify_POST`	Un usuario envía un código MFA enviado a su número de teléfono.
`mfa_phone_resendCode_POST`	Un usuario envía una solicitud para volver a enviar un código MFA a su número de teléfono.
`mfa_totp_POST`	Un usuario envía un código MFA TOTP.
`signup_POST`	Un usuario envía la información a su página de inicio de sesión administrado `/signup`.
`signup_confirm_POST`	Un usuario envía un código de confirmación a través de un correo electrónico o un mensaje SMS.
`verifyCode_POST`	Un usuario envía una contraseña de un solo uso (OTP) para la autenticación sin contraseña.
`passkeys_add_POST`	Un usuario envía una solicitud para registrar una nueva credencial de clave de acceso.
`passkeys_add_GET`	Un usuario navega a la página en la que puede registrar una clave de acceso.
`login_passkey_POST`	Un usuario inicia sesión con una clave de acceso.

> **Note**
>
> Amazon Cognito registra las solicitudes específicas de un usuario, `UserSub` pero no `UserName` en los CloudTrail registros. Si desea buscar un usuario para un `UserSub` determinado, llame a la API de `ListUsers` y utilice un filtro para sub.

Eventos de grupos de identidades

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.

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
```

Ejemplo de evento de Amazon Cognito

Amazon Cognito registra información AWS CloudTrail sobre la actividad de autenticación de usuarios y la actividad de gestión administrativa. Esto se aplica tanto a los grupos de usuarios como a los grupos de identidades. Por ejemplo, puede ver los eventos `GetId` y `UpdateIdentityPool` del mismo registro o los eventos `UpdateAuthEventFeedback` y `SetRiskConfiguration`. También verá los registros de grupos de usuarios de la actividad de la interfaz de usuario alojada que no se correspondan con las operaciones de la API de grupos de usuarios. En esta sección se muestran algunos ejemplos de registros que puede ver. Para comprender el esquema de CloudTrail eventos de cualquier operación, genere una solicitud para esa operación y revise los eventos que crea en su registro.

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
- Ejemplo de CloudTrail evento para una solicitud de SAML
- Ejemplos de CloudTrail eventos para solicitudes al punto final del token
- Ejemplo de CloudTrail evento para CreateIdentityPool
- Ejemplo de CloudTrail evento para GetCredentialsForIdentity
- Ejemplo de CloudTrail evento para GetId
- Ejemplo de CloudTrail evento para GetOpenIdToken
- Ejemplo de CloudTrail evento para GetOpenIdTokenForDeveloperIdentity
- Ejemplo de CloudTrail evento para UnlinkIdentity

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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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 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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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 back-end envía una solicitud `client_credentials` de un token de acceso a su 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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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.auth.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_EXAMPLE"
    },
    "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"
}
```


Acceda a Amazon Cognito mediante un punto final de interfaz (AWS PrivateLink)

Puede utilizarla AWS PrivateLink para crear una conexión privada entre su VPC y Amazon Cognito. Puede acceder a Amazon Cognito como si estuviera en su VPC, sin utilizar una puerta de enlace a

Internet, un dispositivo NAT, una conexión VPN o una conexión. Direct Connect Las instancias de su VPC no necesitan direcciones IP públicas para acceder a Amazon Cognito.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a Amazon Cognito.

Para obtener más información, consulte Acceso a los Servicios de AWS a través de AWS PrivateLink en la Guía de AWS PrivateLink .

> **Important**
>
> Actualmente, no se admiten los siguientes tipos de autenticación: AWS PrivateLink
>
> 1. La autorización de máquina a máquina (M2M) con las credenciales del cliente de OAuth 2.0 fluye desde el punto final del token. Se admite la autorización M2M con la operación de `GetClientToken` API, ya que no requiere un dominio de grupo de usuarios.
> 2. Sign-in con inicio de sesión gestionado y la clásica interfaz de usuario alojada.

Temas

- Flujos de autenticación para AWS PrivateLink integración
- Modos operativos para AWS PrivateLink
- Consideraciones sobre Amazon Cognito
- Controlar el acceso con políticas de control de recursos
- Cree un punto final de interfaz para Amazon Cognito
- Creación de una política de puntos de conexión para el punto de conexión de interfaz
- Cree una política basada en la identidad para AWS PrivateLink operaciones

Flujos de autenticación para AWS PrivateLink integración

En la siguiente tabla se describen los flujos de autenticación disponibles para los clientes en las VPC y las políticas de IAM que puede aplicar para gobernarlos. Las políticas que puede evaluar en las solicitudes a los grupos de usuarios son las políticas de control de recursos (RCP), las políticas de puntos finales de VPC y las políticas basadas en la identidad.

Recurso	Flujo de autenticación	Políticas evaluadas cuando el cliente transita por un punto final de VPC	Las políticas se evalúan cuando el origen del cliente es público
Grupo de usuarios	Inicio de sesión gestionado e inicio de sesión clásico en la interfaz de usuario alojada	Ninguno (sin acceso) [1]	Ninguna [2]
Grupo de usuarios	Machine-to-machine autorización desde el punto final del token (la autorización M2M con la operación de `GetClientToken` API está cubierta por la fila de solicitudes no autenticadas del SDK y la API REST)	Ninguno (sin acceso) [1]	Ninguna [2]
Grupo de usuarios	Solicitudes no autenticadas del SDK y la API REST	Políticas de puntos finales de RCP y VPC [3]	RCP
Grupo de usuarios	Solicitudes autenticadas SigV4 de SDK y API REST	RCP, políticas de puntos finales de VPC, políticas basadas en la identidad [3]	RCP, políticas basadas en la identidad
Grupo de identidades	Solicitudes no autenticadas del SDK y la API REST (flujos básicos y mejorados)	Políticas de puntos finales de RCP y VPC	RCP
Grupo de identidades	Solicitudes autenticadas SigV4 de SDK y API REST (flujo autenticado por el desarrollador)	RCP, políticas basadas en la identidad	RCP, políticas basadas en la identidad

[1] Los puntos finales de VPC no aceptan solicitudes de dominios de grupos de usuarios. Si el cliente tiene una ruta a Internet, se aplica la NAT, lo que hace que el origen sea público.

[2] La existencia de un dominio de grupo de usuarios impide completar cualquier solicitud de grupo de usuarios que transite por un punto final de VPC. Cualquier cliente puede tomar rutas de transporte público solo hasta el dominio del grupo de usuarios y los puntos finales del servicio de API, lo que hace que el punto final de la VPC no pueda utilizarse para el grupo de usuarios. Los grupos de usuarios con dominios asignados son incompatibles con. AWS PrivateLink

[3] El grupo de usuarios no debe tener un dominio asignado.

Modos operativos para AWS PrivateLink

Los siguientes modelos de implementación de ejemplo son compatibles con AWS PrivateLink Amazon Cognito.

Recurso	Implementación	Acciones
Grupo de usuarios	Aplicación de API REST o SDK totalmente privada	1. Eliminación de un dominio 2. Creación de un punto de conexión de VPC 3. Configure el RCP para `Deny` todas las acciones de cognito-idp, excepto las de la VPC
Grupo de usuarios	Pública y privada	1. Eliminación de un dominio 2. Creación de un punto de conexión de VPC
Grupo de usuarios	Servidor de autorización OAuth 2.0 público o privado	1. No disponible para VPC
Grupo de identidades	Totalmente privado	1. Creación de un punto de conexión de VPC 2. Configure el RCP para `Deny` todas las acciones de identidad cognitiva, excepto las de la VPC
Grupo de identidades	Pública y privada	1. Creación de un punto de conexión de VPC

Consideraciones sobre Amazon Cognito

Antes de configurar un punto final de interfaz para Amazon Cognito, consulte las consideraciones de la AWS PrivateLink guía. Amazon Cognito permite realizar llamadas a todas las acciones de la API de Amazon Cognito a través del punto de enlace de la interfaz. Para obtener más información sobre estas operaciones, consulte la referencia de la API de grupos de usuarios de Amazon Cognito y la referencia de la API de identidades federadas de Amazon Cognito.

Temas

- Grupos de usuarios y AWS PrivateLink
- Grupos de identidades y AWS PrivateLink

Grupos de usuarios y AWS PrivateLink

Puede realizar solicitudes a todos los grupos de usuarios (operaciones de API) a través del punto final de la interfaz, pero no a las operaciones que su aplicación solicite al grupo de usuarios (servidor de autorización de OAuth 2.0), por ejemplo, la concesión de credenciales de cliente y el inicio de sesión gestionado.

La API `cognito-idp` de grupos de usuarios tiene operaciones de API no autenticadas, autenticadas y autorizadas por token. Puedes conceder permisos para operaciones autenticadas en las políticas de control de recursos y puntos finales de la VPC. También puedes conceder permisos para operaciones no autenticadas y autorizadas por token, a diferencia de lo que ocurre con las políticas basadas en la identidad. Los tipos de políticas de punto final y control de recursos de VPC pueden evaluar y denegar o permitir solicitudes de operaciones que de otro modo serían públicas.

Las solicitudes a los puntos de enlace del dominio también son públicas, pero no se pueden evaluar en las políticas. El DNS privado de VPC no enruta las solicitudes de dominios de grupos de usuarios a su punto final de VPC. Solo puedes realizar solicitudes de servicios de dominio a través de rutas de Internet públicas. Para obtener más información, consulte Efectos de las políticas en las operaciones de los grupos de usuarios.

Temas

- Operaciones admitidas
- Efectos de las políticas en las operaciones de los grupos de usuarios

Operaciones admitidas

Los sistemas de una VPC pueden enviar solicitudes a las acciones de la API del grupo de usuarios, pero no a los puntos finales del dominio del grupo de usuarios. Los flujos de trabajo de OpenID Connect (OIDC) y OAuth 2.0 que utilizan puntos de enlace de dominio, por ejemplo, máquina a máquina (M2M), inicio de sesión federado y concesiones de códigos de autorización, son inaccesibles a través de los puntos de enlace de VPC. Las políticas de punto final de VPC no afectan a estos flujos de trabajo HTTP y no pueden procesarlos. Las solicitudes a los puntos de enlace de dominio desde una VPC siempre fallan en el punto de enlace de la interfaz, pero siguen estando disponibles a través del DNS público y el enrutamiento al configurar los puntos de enlace de la VPC para sus grupos de usuarios.

Para evitar la asignación de dominios desde los sistemas de una VPC, Amazon Cognito `CreateUserPoolDomain` bloquea las solicitudes en el punto final de la interfaz. Esto evita que se agreguen dominios a sus grupos de usuarios desde sistemas que se encuentran en una VPC. Para evitar que se añada un dominio de todos los sistemas, aplique a su sistema una política de control de recursos (RCP) como la que se muestra en el siguiente ejemplo. Cuenta de AWS Esta política bloquea la `CreateUserPoolDomain` acción contra el grupo de usuarios especificado.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Principal": "*",
            "Effect": "Deny",
            "Action": [
                "cognito-idp:CreateUserPoolDomain"
            ],
            "Resource": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE"
        }
    ]
}
```

Es posible que su grupo de usuarios tenga un dominio y, en todos los casos, ese dominio no estará disponible a través de él AWS PrivateLink. Todas las solicitudes de API de grupos de SDK-based usuarios a puntos finales de `cognito-idp` servicio aceptan solicitudes AWS PrivateLink directas, con la excepción de`CreateUserPoolDomain`. Los puntos finales del servicio de API del grupo de usuarios y los puntos finales del dominio permanecen siempre accesibles a través de rutas públicas de Internet. Para abordar el acceso desde fuentes públicas, implemente AWS WAF las ACL web.

Efectos de las políticas en las operaciones de los grupos de usuarios

Todas las operaciones de la API del grupo de usuarios, incluso las que suelen ser públicas y no autenticadas, se pueden controlar mediante políticas de puntos finales de VPC y políticas de control de recursos (RCP). También puedes aplicar restricciones al acceso al grupo de usuarios en políticas basadas en identidad con claves de condición de VPC. En las políticas basadas en la identidad, solo las solicitudes que incluyen información de autenticación en formato SigV4 se pueden controlar. El inicio de sesión gestionado y las operaciones clásicas de interfaz de usuario alojada son una categoría independiente y no son aptas para el tránsito de VPC ni para la aplicación de ningún tipo de política a sus acciones.

Operaciones no autenticadas

Las operaciones de Amazon Cognito para las aplicaciones del lado del cliente no se autentican con SigV4. Las operaciones de ejemplo se encuentran en la política de ejemplo en. Creación de una política de puntos de conexión para el punto de conexión de interfaz Otros ejemplos de operaciones no autenticadas son `GetUser` y. `AssociateSoftwareToken` Cuando agrega estas operaciones a las políticas basadas en la identidad, no tienen ningún efecto. Sin embargo, puede permitir o restringir el acceso a estas acciones en las políticas de puntos finales de la VPC y los RCP.

Las operaciones no autenticadas no están asociadas a una entidad principal de IAM. Su política de puntos finales de VPC o RCP debe permitir que todos los principales realicen estas acciones.

Operaciones autenticadas

Las operaciones de la API para la administración del grupo de usuarios y la autenticación del lado del servidor se autentican con SigV4. Para las operaciones autenticadas, puede restringir los principales con políticas de punto final que aplique al punto final de la VPC, políticas de control de recursos en su organización y políticas basadas en identidad que aplique a los principales. Identity-based y las políticas de control de recursos utilizan claves de condición basadas en la red, como y. VPC-aware `aws:SourceVpc` `aws:SourceVpce`

Para obtener más información sobre las clases de operaciones de API del lado del servidor, del lado del cliente y administrativas para los grupos de usuarios, consulte. Modelos de autorización para la autenticación de API y SDK

Grupos de identidades y AWS PrivateLink

Los grupos de identidades de Amazon Cognito admiten todas las operaciones de la API de principio a fin. AWS PrivateLink

Temas

- Operaciones admitidas
- Limitaciones del contexto de la red con AWS STS integración
- Service-specific claves de contexto

Operaciones admitidas

Todas las operaciones de la API de los grupos de identidades se admiten a través del punto final de la interfaz. Los grupos de identidades no tienen puntos de enlace de dominio y no están sujetos a las mismas limitaciones. Sin embargo, los grupos de identidades tienen consideraciones específicas para los controles de acceso basados en la red debido a su integración con. AWS STS

Limitaciones del contexto de la red con AWS STS integración

Los grupos de identidades utilizan AWS STS `AssumeRoleWithWebIdentity` operaciones para proporcionar AWS credenciales temporales. Cuando los grupos de identidades AWS STS intervienen AWS PrivateLink en el flujo de autenticación mejorada, las claves de contexto de red`aws:SourceIp`, como`aws:SourceVpc`, y `aws:SourceVpce` contienen valores de la infraestructura de servicios de los grupos de identidades, no del contexto de red de la aplicación.

Si su función de IAM, las políticas de confianza o las políticas de control de recursos (RCP) utilizan claves de condición basadas en la red para restringir el acceso, es posible que las operaciones de los grupos de identidades se denieguen inesperadamente. Para abordar esta limitación, puede utilizar uno de los siguientes enfoques:

Etiquetas principales para la identificación del servicio

Etiquete las funciones de IAM que se utilizan en los grupos de identidades y modifique sus políticas para permitir las operaciones cuando el principal tenga la etiqueta adecuada. En primer lugar, añada una etiqueta a su rol de grupo de identidades:

```
aws iam tag-role \
    --role-name MyIdentityPoolRole \
    --tags Key=CognitoServiceCall,Value=true
```

A continuación, modifique las políticas basadas en la red para permitir etiquetar a los principales. Por ejemplo, en un RCP:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Resource": "*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": ["allowed-ip-ranges"]
                },
                "StringNotEqualsIfExists": {
                    "aws:ResourceTag/CognitoServiceCall": "true"
                }
            }
        }
    ]
}
```

Service-specific claves de contexto

Los grupos de identidades proporcionan claves de contexto específicas del servicio para la autorización a nivel de recursos en las políticas de puntos finales de VPC y los RCP. Con estas claves de contexto, puede habilitar un control de acceso detallado y distinguir entre usuarios autenticados y no autenticados en las políticas.

Claves de contexto específicas del servicio disponibles para operaciones ajenas al SIGv4, como GetId, GetCredentialsForIdentity, GetOpenIdToken, UnlinkIdentity

- `cognito-identity-unauth:IdentityPoolArn`- Filtra el acceso por el ARN del grupo de identidades para los usuarios no autenticados
- `cognito-identity-unauth:AccountId`- Filtra el acceso por Cuenta de AWS ID para los usuarios no autenticados
- `cognito-identity-auth:IdentityPoolArn`- Filtra el acceso por el ARN del grupo de identidades para los usuarios autenticados
- `cognito-identity-auth:AccountId`- Filtra el acceso por el Cuenta de AWS ID de los usuarios autenticados

Claves de contexto específicas del servicio disponibles para las operaciones de SigV4, como DeleteIdentities and DescribeIdentity

- `cognito-identity:IdentityPoolArn`- Filtra el acceso por el ARN del grupo de identidades

Puedes usar estas claves de contexto en las políticas de puntos finales de la VPC para restringir el acceso en función del estado de la autenticación, como se muestra en el siguiente ejemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "cognito-identity:GetId",
                "cognito-identity:GetCredentialsForIdentity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cognito-identity-unauth:IdentityPoolArn": "arn:aws:cognito-
identity:us-east-1:123456789012:identitypool/us-east-1:12345678-ffff-ffff-ffff-123456"
                }
            }
        }
    ]
}
```

Controlar el acceso con políticas de control de recursos

Amazon Cognito permite controlar el acceso a sus recursos con políticas de control de recursos (RCP). Con las claves de condición basadas en la red, los RCP pueden definir las redes y las acciones que están permitidas para AWS PrivateLink acceder a sus grupos de usuarios y grupos de identidades. Las `Action` instrucciones de los RCP pueden controlar el acceso a las operaciones de API de grupos de usuarios autenticados y no autenticados.

Por ejemplo, la siguiente política de ejemplo impide el acceso a todos los grupos de usuarios desde una VPC específica.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCognitoAccessOutsideVPC",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "cognito-idp:*",
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceVpc": "vpc-02d6770f46ef1653b"
        }
      }
    }
  ]
}
```

Cree un punto final de interfaz para Amazon Cognito

Puede crear un punto final de interfaz para Amazon Cognito mediante la consola de Amazon VPC o el (). AWS Command Line Interface AWS CLI Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

Cree un punto final de interfaz para los grupos de usuarios de Amazon Cognito con el siguiente nombre de servicio:

```
com.amazonaws.region.cognito-idp
```

Cree un punto final de interfaz para los grupos de identidades de Amazon Cognito con el siguiente nombre de servicio:

```
com.amazonaws.region.cognito-identity
```

Si habilita el DNS privado para el punto final de la interfaz, puede realizar solicitudes de API a Amazon Cognito con su nombre de DNS regional predeterminado. Por ejemplo, `cognito-idp.us-east-1.amazonaws.com` para grupos de usuarios y grupos `cognito-identity.us-east-1.amazonaws.com` de identidades.

Creación de una política de puntos de conexión para el punto de conexión de interfaz

Una política de punto de conexión es un recurso de IAM que puede adjuntar al punto de conexión de su interfaz. La política de puntos de conexión predeterminada permite el acceso total a Amazon Cognito a través del punto de enlace de la interfaz. Para controlar el acceso permitido a Amazon Cognito desde su VPC, adjunte una política de punto de conexión personalizada al punto de enlace de la interfaz.

Una política de punto de conexión especifica la siguiente información:

- Las entidades principales que pueden llevar a cabo acciones (Cuentas de AWS, usuarios de IAM y roles de IAM).
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.
- Las condiciones que deben cumplirse antes de permitir o denegar la solicitud.

Para obtener más información, consulte Control del acceso a los servicios con políticas de punto de conexión en la Guía del usuario de AWS PrivateLink .

Ejemplo: política de puntos finales de VPC para acciones de grupos de usuarios

El siguiente es un ejemplo de una política de puntos finales personalizada para grupos de usuarios. Al adjuntar esta política al punto final de la interfaz, concede acceso a las acciones del grupo de usuarios enumeradas a todos los principales de todos los recursos.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:assumed-role/MyWebAppRole/
MyWebAppSession"
         },
         "Effect": "Allow",
         "Action": [
            "cognito-idp:AdminInitiateAuth",
            "cognito-idp:AdminRespondToAuthChallenge",
            "cognito-idp:AdminSetUserPassword"
```

```
         ],
         "Resource":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE"
      },
      {
         "Effect": "Allow",
         "Action": [
            "cognito-idp:InitiateAuth",
            "cognito-idp:RespondToAuthChallenge",
            "cognito-idp:ForgotPassword",
            "cognito-idp:ConfirmForgotPassword"
         ],
         "Resource":"arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE"
      }
   ]
}
```

Ejemplo: política de puntos finales de VPC para acciones de grupos de identidades

El siguiente es un ejemplo de una política de puntos finales personalizada para grupos de identidades. Esta política utiliza claves de contexto específicas del servicio para restringir el acceso a los usuarios autenticados de un grupo de identidades específico.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": [
            "cognito-identity:GetId",
            "cognito-identity:GetCredentialsForIdentity",
            "cognito-identity:GetOpenIdToken"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "cognito-identity-auth:IdentityPoolArn": "arn:aws:cognito-identity:us-
east-1:123456789012:identitypool/us-east-1:12345678-ffff-ffff-ffff-123456"
            }
         }
      }
```

```
    ]
}
```

Cree una política basada en la identidad para AWS PrivateLink operaciones

Identity-basedlas políticas son recursos de IAM que puede adjuntar a los directores. AWS Puede controlar el acceso a Amazon Cognito a través de puntos de enlace de VPC con políticas de operaciones basadas en la identidad. IAM-authenticated A diferencia de las políticas de puntos finales, en las políticas basadas en la identidad no se pueden configurar los permisos para las operaciones no autenticadas. Las operaciones autenticadas o administrativas requieren la autorización de la versión 4 firmada. En el caso de los grupos de usuarios, las operaciones autenticadas incluyen tanto las solicitudes de autenticación del lado del servidor como las solicitudes AdminInitiateAuthadministrativas. UpdateUserPool En el caso de los grupos de identidades, las operaciones autenticadas incluyen solicitudes administrativas como y. DeleteIdentitiesDescribeIdentity

Una política basada en la identidad especifica la siguiente información:

- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.
- Las condiciones que deben cumplirse antes de permitir o denegar la solicitud.

Ejemplo: política basada en la identidad para la autenticación de grupos de usuarios en el servidor

El siguiente ejemplo de política otorga acceso a las acciones del grupo de usuarios enumeradas en el grupo de usuarios especificado, desde el punto final especificado. Aplique esta política a la función de IAM asumida para su aplicación web.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-idp:AdminInitiateAuth",
                "cognito-idp:AdminRespondToAuthChallenge",
                "cognito-idp:AdminSetUserPassword"
            ],
            "Resource": "arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-
east-1_EXAMPLE",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpce": "vpce-1a2b3c4d"
                }
            }
        }
    ]
}
```

Ejemplo: política basada en la identidad para las operaciones administrativas del grupo de identidades

El siguiente ejemplo de política otorga acceso a las acciones administrativas del grupo de identidades desde el punto de enlace de VPC especificado. Aplique esta política al director de IAM que debe administrar el grupo de identidades.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:DeleteIdentities",
                "cognito-identity:DescribeIdentity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpce": "vpce-1a2b3c4d",
                    "cognito-identity:IdentityPoolArn": "arn:aws:cognito-identity:us-
east-1:123456789012:identitypool/us-east-1:12345678-ffff-ffff-ffff-123456"
                }
            }
        }
    ]
}
```

Validación de la conformidad para Amazon Cognito

Third-party los auditores 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. AWS HIPAA-compliant
- AWS recursos AWS de cumplimiento
- 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 CSPM— 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

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 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, Amazon Cognito envía los eventos a los proyectos de Amazon Pinpoint dentro de la misma región de Amazon Pinpoint. Si Amazon Pinpoint no admite una región, Amazon Cognito solo admite 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.
- Paquetes de cifrado con perfecto secreto directo (PFS), como el DHE (Ephemeral) o el ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Configuración y análisis de vulnerabilidades en grupos de usuarios de Amazon Cognito

AWS se encarga de 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 de 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 de 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.

AWS políticas de IAM gestionadas que otorgan 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 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.

AWS políticas de IAM gestionadas que Amazon Cognito concede a los usuarios invitados

- `AmazonCognitoUnAuthedIdentitiesSessionPolicy`: en combinación con una política de sesión en línea, limita los permisos que los administradores de IAM pueden conceder a los usuarios invitados del grupo de identidades. Amazon Cognito aplica automáticamente esta política a las sesiones de invitados. Para obtener más información, consulte La AWS política de sesiones gestionadas para invitados.

Amazon Cognito se actualiza a AWS políticas administradas

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
`AmazonCognitoPowerUser` : cambio	Amazon Cognito agregó nuevas acciones para permitir	27 de febrero de 2025

Cambio	Descripción	Fecha
	el uso de la operación de AWS End User Messaging SMS API DescribeAccountAttributespara grupos de usuarios avanzados administrativos de Amazon Cognito.	
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` : cambio	Amazon Cognito agregó nuevas acciones para permitir el uso de AWS Key Management Service usuarios no autenticados (invitados) en grupos de identidades.	30 de octubre de 2024
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` : cambio	Amazon Cognito ha agregado nuevas acciones para permitir el uso de Amazon Location Service a usuarios no autenticados (invitados) en grupos de identidades.	9 de agosto de 2024
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` : nueva política	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` : cambio	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` : cambio	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` : cambio	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

Legacy

> **Important**
>
> Amazon Cognito Sync está cerrado a nuevos clientes. Si va a iniciar una aplicación nueva, utilice en su lugar una alternativa actual, como AWS AppSyncDynamoDB. El contenido de esta sección se conserva para los clientes con implementaciones de Amazon Cognito Sync existentes.

En esta sección se recopilan las funciones de Amazon Cognito de generaciones anteriores. Amazon Cognito Sync está cerrado a nuevos clientes, como se ha indicado anteriormente. La interfaz de usuario alojada clásica es una versión de marca anterior que aún puede seleccionar por cliente de aplicación, pero el inicio de sesión administrado es la versión de marca actual para nuevos trabajos; para mover un cliente de aplicación de la interfaz de usuario alojada clásica al inicio de sesión administrado, consulteMigración de la clásica interfaz de usuario alojada al inicio de sesión gestionado.

Temas

- Amazon Cognito Sync
- Uso de Amazon Cognito Sync con grupos de identidades
- Personalización de marca de IU alojada (clásica)

Amazon Cognito Sync

> **Note**
>
> Amazon Cognito Sync ya no está abierto a nuevos clientes. Para ver alternativas a Amazon Cognito Sync, explore AWS AppSyncDynamoDB. Más información.

> **Note**
>
> Si es la primera vez que utiliza Amazon Cognito Sync, utilice AWS AppSync. Al igual que Amazon Cognito Sync, AWS AppSync es un servicio para sincronizar los datos de las aplicaciones en todos los 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 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 del perfil de usuario entre dispositivos móviles y la web sin utilizar su propio back-end. 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

- Cambio de disponibilidad de Amazon Cognito Sync
- Introducción a Amazon Cognito Sync
- Sincronización de datos entre clientes
- Gestión de las devoluciones de llamada de eventos
- Implementación de la sincronización mediante inserción
- Implementación de flujos de Amazon Cognito Sync
- Personalización de los flujos de trabajo con Amazon Cognito Events

Cambio de disponibilidad de Amazon Cognito Sync

Amazon Cognito Sync ya no está abierto a nuevos clientes. Para ver alternativas a Amazon Cognito Sync, explore AWS AppSyncDynamoDB.

Esta página proporciona información sobre los cambios de Amazon Cognito Sync e incluye alternativas para los clientes de Amazon Cognito Sync.

AWS AppSynces un servicio sin servidor que permite a los desarrolladores crear y gestionar las API de GraphQL. DynamoDB es un almacén de valores clave para datos por usuario, como perfiles,

preferencias y configuraciones de aplicaciones. La migración desde Amazon Cognito Sync requiere que el nuevo servicio transfiera datos y actualice las aplicaciones a las diferentes API y patrones de consulta.

Si tiene más preguntas, póngase en contacto con AWS Support.

Preguntas frecuentes

¿Qué significa esto para el servicio?

Amazon Cognito Sync ya no aceptará nuevos clientes. El servicio seguirá funcionando para los clientes actuales, pero no se desarrollarán nuevas funciones.

¿Cómo se verán afectados los clientes actuales?

Los clientes actuales no sufrirán ninguna interrupción en sus cargas de trabajo. Puede seguir utilizando Amazon Cognito Sync con normalidad.

¿Cuáles son las alternativas recomendadas a Amazon Cognito Sync?

- AWS AppSync— Para la sincronización de datos en tiempo real entre dispositivos con las API de GraphQL.
- DynamoDB: para almacenar de forma sencilla los datos de usuario con valores clave (perfiles, preferencias, ajustes).

¿Cómo puedo migrar mis datos?

La migración requiere actualizar el esquema de datos, migrar los datos de la tienda Amazon Cognito Sync a la alternativa elegida y actualizar las aplicaciones cliente para usar los nuevos servicios.

¿Cómo puedo obtener ayuda si tengo problemas?

Si tiene alguna duda al respecto, póngase en contacto con AWS Support.

Introducción a Amazon Cognito Sync

> **Note**
>
> Si es la primera vez que usa Amazon Cognito Sync, utilice AWS AppSync. Al igual que Amazon Cognito Sync, AWS AppSync es un servicio para sincronizar los datos de las aplicaciones en todos los 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.

> **Note**
>
> Los ejemplos de código de esta sección utilizan el cliente Amazon Cognito Sync del SDK AWS móvil archivado (para Android, iOS, Unity y Xamarin/.NET). Ese SDK móvil ya no se mantiene y sus paquetes no están disponibles en los repositorios de paquetes actuales. Los ejemplos se conservan para los clientes actuales que siguen integrándose con ese SDK. Para un nuevo desarrollo, utilice AWS AppSync.

Amazon Cognito Sync es una biblioteca de AWS servicios y clientes que permite la sincronización entre dispositivos de los datos de usuario relacionados con las aplicaciones. 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 entre clientes.

Sincronización de datos entre clientes

> **Note**
>
> Si es la primera vez que utiliza Amazon Cognito Sync, utilice AWS AppSync. Al igual que Amazon Cognito Sync, AWS AppSync es un servicio para sincronizar los datos de las aplicaciones en todos los 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 adquiere AWS credenciales temporales para que su aplicación pueda acceder a sus AWS recursos. 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 lo estás usando CocoaPods, `<AWSiOSSDKv2/AWSCore.h>` sustitúyelo 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 Amazon Cognito Sync Manager 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 `synchronizeOnConnectivity()` se llama varias veces, solo se conserva la última solicitud de sincronización y solo se activa 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 las devoluciones de llamada de eventos.

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 las devoluciones de llamada de eventos.

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 las devoluciones de llamada de eventos.

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 las devoluciones de llamada de eventos.

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();
```

Synchronize se ejecuta de forma asíncrona y termina llamando a una de las diversas devoluciones de llamada 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 las devoluciones de llamada de eventos.

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 las devoluciones de llamada de eventos.

Gestión de las devoluciones de llamada de eventos

> **Note**
>
> Si es la primera vez que utiliza Amazon Cognito Sync, utilice AWS AppSync. Al igual que Amazon Cognito Sync, AWS AppSync es un servicio para sincronizar los datos de las aplicaciones en todos los 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.

Como desarrollador de Amazon Cognito Sync, puede implementar varias devoluciones de llamada para gestionar diferentes eventos y escenarios de sincronización. La interfaz de `SyncCallback` del SDK de Android configura las notificaciones sobre la sincronización de conjuntos de datos, incluso `onSuccess()` cuando un conjunto de datos se descarga correctamente, `onFailure()` cuando se produce una excepción y `onConflict()` para resolver conflictos entre los datos locales y remotos.

En el SDK de iOS, puede registrarse para recibir notificaciones similares como `AWSCognitoDidStartSynchronizeNotification` y establecer controladores como `AWSCognitoRecordConflictHandler` para resolver conflictos. Las JavaScript plataformas Unity y Xamarin tienen mecanismos de devolución de llamadas análogos. Al implementar estas

devoluciones de llamada, la aplicación puede gestionar sin problemas los distintos eventos y escenarios de sincronización que se pueden producir al utilizar Amazon Cognito Sync.

Android

SyncCallback Interfaz

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;
}
```

en DatasetDeleted

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;
```

```
}
```

en DatasetMerged

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 emite una serie de `NSNotification` eventos durante una llamada sincronizada. 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. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando.

AWSCognitoDidEndSynchronizeNotification

Se llama cuando finaliza una operación de sincronización (correctamente o no). `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando.

AWSCognitoDidFailToSynchronizeNotification

Se llama cuando una operación de sincronización falla. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando, y el error clave, que contiene el error que provocó el error.

AWSCognitoDidChangeRemoteValueNotification

Se llama cuando los cambios locales se envían de forma correcta a Amazon Cognito. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando, y las claves clave, que contienen un NSArray con las claves de registro que se insertaron.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Se llama cuando un valor local cambia debido a una operación de sincronización. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando, y las claves clave, que contienen un NSArray de claves de registro que se han modificado.

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.

Al implementar y asignar una, AWSCognitoRecordConflictHandler puede modificar la resolución de conflictos predeterminada. El conflicto AWSCognitoConflict de parámetros de entrada contiene un AWSCognitoRecord objeto tanto para los datos en caché local como para el registro conflictivo del almacén de sincronización. AWSCognitoConflict Si lo usa, puede resolver el conflicto con el registro local: [resolución de conflictosWithLocalRecord], el registro remoto: [resolución de conflictosWithRemoteRecord] o con un valor completamente nuevo: [resolución WithValue:value de conflictos]. Si se devuelve cero con este método, se impide que la sincronización continúe y los conflictos se vuelven a presentar la próxima 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 `AWSCognitoDatasetDeletedHandler` se implementa la opción no, los datos locales se purgan 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 controlador recibe el nombre del conjunto de datos raíz, así como una

matriz de nombres de conjuntos de datos que se marcan como combinaciones del conjunto de datos raíz.

Si `DatasetMergeHandler` se implementa la opción no, estos conjuntos de datos se ignoran, pero siguen ocupando espacio en el total máximo de 20 conjuntos de datos de la identidad.

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 emite una serie de `NSNotification` eventos durante una llamada sincronizada. 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. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando.

AWSCognitoDidEndSynchronizeNotification

Se llama cuando finaliza una operación de sincronización (correctamente o no). `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando.

AWSCognitoDidFailToSynchronizeNotification

Se llama cuando una operación de sincronización falla. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando, y el error clave, que contiene el error que provocó el error.

AWSCognitoDidChangeRemoteValueNotification

Se llama cuando los cambios locales se envían de forma correcta a Amazon Cognito. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando, y las claves clave, que contienen un NSArray con las claves de registro que se insertaron.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Se llama cuando un valor local cambia debido a una operación de sincronización. `userInfo`Contiene el conjunto de datos clave, que es el nombre del conjunto de datos que se está sincronizando, y las claves clave, que contienen un NSArray de claves de registro que se han modificado.

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. `AWSCognitoConflict`Si lo usa, puede resolver el conflicto con el registro local: [resolución de conflictosWithLocalRecord], el registro remoto: [resolución de conflictosWithRemoteRecord] o un valor completamente nuevo: [resolución de conflictos]. WithValue:value Si se devuelve cero con este método, se impide que la sincronización continúe y los conflictos se vuelven a presentar la próxima 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 `AWSCognitoDatasetDeletedHandler` se implementa la opción no, los datos locales se purgan 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 controlador recibe el nombre del conjunto de datos raíz, así como una matriz de nombres de conjuntos de datos que se marcan como combinaciones del conjunto de datos raíz.

Si `DatasetMergeHandler` se implementa la opción no, estos conjuntos de datos se ignoran, pero siguen ocupando espacio en el total máximo de 20 conjuntos de datos de la identidad.

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);

}
```

no () DatasetDeleted

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, el conjunto de datos no se elimina.

```
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);

}
```

en DatasetMerged ()

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, puedes configurar diferentes callbacks para que se activen al usar 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, el conjunto de datos no se elimina.

```
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, puedes configurar diferentes devoluciones de llamada que se activan al usar 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, el conjunto de datos no se elimina.

```
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;
}
```

Implementación de la sincronización mediante inserción

> **Note**
>
> Si es la primera vez que utiliza Amazon Cognito Sync, utilice AWS AppSync. Al igual que Amazon Cognito Sync, AWS AppSync es un servicio para sincronizar los datos de las aplicaciones en todos los 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 automática no es compatible con Unity ni Xamarin. JavaScript

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 AWS Identity and Access Management consola, configure sus funciones de IAM para tener acceso total a Amazon SNS o cree una nueva función que tenga acceso total a 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`.

JSON

```
{
    "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

La aplicación debe 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 tu implementación del BroadcastReceiver objeto de Android, puedes comprobar la última versión del conjunto de datos modificado y decidir si tu aplicación necesita volver a sincronizarse:

```
@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. Esto está disponible por el bien de la OrCreateDataset convocatoria abierta.
- `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.

Usa la sincronización push en tu 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 limitado de los APN; en el modo de liberación, 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 limitado de los APN; en el modo de liberación, 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.

Implementación de flujos de Amazon Cognito Sync

> **Note**
>
> Si es la primera vez que utiliza Amazon Cognito Sync, utilice AWS AppSync. Al igual que Amazon Cognito Sync, AWS AppSync es un servicio para sincronizar los datos de las aplicaciones en todos los 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).

Una vez que haya configurado correctamente las transmisiones de Amazon Cognito, todas las actualizaciones posteriores de los conjuntos de datos de este grupo de identidades se envían a la transmisión.

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. Tras iniciar una operación de publicación masiva, ya sea a través de la consola o directamente a través de la API, Amazon Cognito comienza a publicar estos datos en la misma transmisión 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.

Personalización de los flujos de trabajo con Amazon Cognito Events

> **Note**
>
> Si es la primera vez que utiliza Amazon Cognito Sync, utilice AWS AppSync. Al igual que Amazon Cognito Sync, AWS AppSync es un servicio para sincronizar los datos de las aplicaciones en todos los 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 Events le permite ejecutar una AWS Lambda función en respuesta a eventos importantes en 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.

Los pasos que se indican a continuación le guiarán por el proceso de configuración de una función Lambda que se ejecute cada vez que se sincronice 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 Lambda asociada, pero llama `UpdateRecords` con credenciales de AWS cuenta (credenciales de desarrollador), no se invocará su función Lambda.

Crear 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 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 añadir este permiso desde la consola de Lambda (consulte Uso de políticas basadas en recursos para AWS Lambda) o mediante la operación Lambda. AddPermission
> Ejemplo de política Lambda Resource-Based
> 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`.
>
> JSON

```
{
    "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": "arn:aws:lambda:us-
east-1:111122223333:function:MyFunction",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "AWS:SourceArn": "arn:aws:cognito-
identity:us-east-1:111122223333:identitypool/us-
east-1:abcdefg-1234-5678-910a-0e8443553f95"
                }
            }
        }
    ]
}
```

5. En la pantalla de la función Configure, especifique un nombre y una descripción para su función. Deje Runtime establecido en "»Node.js. Deje el código sin cambios para nuestro 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, la función Lambda se ejecuta cada vez que se sincroniza 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 Lambda durante UpdateRecords, la función debe responder en un plazo 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 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 del perfil de usuario entre dispositivos móviles y la web sin utilizar su propio back-end. 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 Implementación de flujos de Amazon Cognito Sync.

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 asigna a su nueva función 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 Implementación de flujos de Amazon Cognito Sync.

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 Personalización de los flujos de trabajo con Amazon Cognito Events.

Para obtener más información al respecto AWS Lambda, consulte. AWS Lambda

Personalización de marca de IU alojada (clásica)

Puede usar la o la Consola de administración de AWS API para especificar la AWS CLI configuración de personalización clásica para la interfaz de usuario alojada. Puede cargar una imagen de logotipo personalizada para que se muestre en la aplicación. También puede aplicar algunas opciones de hojas de estilo en cascada (CSS) en el aspecto de la IU.

Puede personalizar los valores predeterminados de la interfaz de usuario y anular los clientes de aplicación individuales con ajustes específicos. Amazon Cognito aplica la configuración predeterminada a todos los clientes de aplicación que no tienen ajustes en el cliente.

En la consola de Amazon Cognito y en las solicitudes de API, 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 solicitud de `SetUICustomization` API Consola de administración de AWS o la suya devolverá un `request parameters too large` error. 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.

Especificación de un logotipo personalizado en la marca clásica

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 configurar una entrada `ImageFile` SetUICustomizationen la API, convierte el archivo en una cadena de Base64-encoded texto o, en el AWS CLI, proporciona una ruta de archivo y deja que Amazon Cognito la codifique por ti.

Especificación de personalizaciones con CSS en la creación clásica de marca

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.
- color es el color del texto. Debe ser un valor de color CSS legal.

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.
- color es el color del texto. Debe ser un valor de color CSS legal.

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.
- El color es el color del texto de los encabezados de las secciones del IdP, por ejemplo, Iniciar sesión con su ID corporativo. Debe ser un valor de color CSS legal.

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.
- El color de fondo es el color del fondo de los registros con secciones transparentes. Debe ser un valor de color CSS legal.

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.

Personalización de la interfaz de usuario alojada con la marca clásica en Consola de administración de AWS

Puede usar el Consola de administración de AWS para especificar la configuración de personalización de la interfaz de usuario para su 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. Cree un dominio desde la pestaña Dominio o actualice su dominio actual. En la Versión de marca, configure su dominio para que utilice la Interfaz de usuario alojada (clásica).
4. Seleccione el menú Inicio de sesión administrado.
5. Para personalizar la configuración de la IU de todos los clientes de aplicación, busque Estilo en Configuración de la interfaz de usuario alojada y seleccione Editar.
6. Para personalizar la configuración de la interfaz de usuario de un cliente de aplicación, vaya al menú Clientes de aplicación y seleccione el cliente de aplicación que desee modificar; luego, busque Estilo de interfaz de usuario alojada (clásico) y seleccione Anular. Seleccione Editar.
7. Para cargar su propio archivo de imagen de logotipo, elija Choose file (Elegir archivo) o bien Replace current file (Reemplazar el archivo actual).
8. 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 reflejan en la interfaz de usuario. Después de personalizar el archivo CSS, elija Choose file (Elegir archivo) o Replace current file (Reemplazar archivo actual) para cargar su archivo CSS personalizado.

Personalizar la interfaz de usuario alojada con la marca clásica en la API de grupos de usuarios y con AWS CLI

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`
- AWS API: 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 la imagen codificada como texto binario en 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>;}"`
- AWS API: SetUICustomization

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
Referencias reorganizadas de API y puntos finales	La referencia a los puntos finales del grupo de usuarios y al inicio de sesión gestionado es ahora una sección de primer nivel de la guía. La guía ya no enlaza con los grupos de usuarios de Amazon Cognito, los grupos de identidades de Amazon Cognito ni las referencias a la API de Amazon Cognito Sync. Estas referencias de API permanecen disponibles en sus respectivas guías de referencia de API.	8 de octubre de 2026
Códigos de estado en las respuestas de bloqueo AWS WAF personalizadas	Amazon Cognito ahora devuelve un código de `403` estado en lugar de los códigos de `5xx` estado de las respuestas de bloqueo AWS WAF personalizadas en el inicio de sesión gestionado y en la clásica interfaz de usuario alojada.	1 de octubre de 2026
Amazon Cognito Sync ya no está abierto a nuevos clientes.	Amazon Cognito Sync ya no está abierto a nuevos clientes.	31 de julio de 2026

	Para ver alternativas a Amazon Cognito Sync, explore AWS AppSync DynamoDB. Más información.	
Machine-to-machine autorización con GetClientToken	Ahora puede emitir tokens de acceso para la autorización de máquina a máquina (M2M) con la operación de la `GetClientToken` API Amazon Cognito, sin un dominio de grupo de usuarios.	29 de julio de 2026
Importación de usuarios con hash de contraseña	Ahora puede importar usuarios con sus códigos hash de contraseña existentes, lo que permite el inicio de sesión inmediato sin necesidad de restablecer la contraseña.	13 de julio de 2026
Administración de límites aprovisionados	Ahora puede gestionar los límites aprovisionados para su grupo de usuarios con las API de aprovisionamiento de Amazon Cognito.	2 de julio de 2026
Amazon Cognito Sync dejará de estar abierto a nuevos clientes a partir del 30 de julio de 2026.	Amazon Cognito Sync dejará de estar abierto a nuevos clientes a partir del 30 de julio de 2026. Los clientes actuales de Amazon Cognito Sync pueden seguir utilizando el servicio. Para ver alternativas a Amazon Cognito Sync, explore AWS AppSync DynamoDB. Más información.	30 de junio de 2026

Selección de versión de TLS para dominios personalizados	Ahora puede seleccionar la versión mínima de TLS para su dominio personalizado al crear o actualizar un dominio de grupo de usuarios con la API de Amazon Cognito.	29 de junio de 2026
Multi-Region replicación para grupos de usuarios.	Ahora puede configurar una copia adicional de sus grupos de usuarios en una versión adicional Región de AWS.	28 de mayo de 2026
Claves KMS administradas por el cliente para el cifrado de grupos de usuarios en reposo.	Ahora puede configurar el cifrado del grupo de usuarios en reposo con su propia clave KMS personalizada.	28 de mayo de 2026
Se actualizó la documentación sobre el comportamiento de reintento del disparador Lambda.	Se actualizó la documentación para aclarar que Amazon Cognito puede volver a intentar las invocaciones de funciones Lambda cuando se agote el tiempo de espera, sin especificar un número fijo de reintentos.	23 de marzo de 2026
Rotación de secretos de clientes para clientes de aplicaciones.	Ahora puedes asociar hasta dos secretos por cliente de aplicación para la rotación de secretos. También puedes proporcionar tu propio valor de secreto personalizado al añadir un secreto.	30 de enero de 2026

Inbound Federation Lambda Trigger está disponible en los grupos de usuarios de Amazon Cognito.	Ahora puede usar el activador lambda de federación entrante para transformar los atributos de los usuarios federados durante el proceso de autenticación con proveedores de identidad externos.	28 de enero de 2026
AWS PrivateLink está disponible en los grupos de identidades de Amazon Cognito.	Ahora puede enviar solicitudes a las operaciones de la API del grupo de identidades de Amazon Cognito a través de puntos de enlace de VPC.	10 de diciembre de 2025
AWS PrivateLink está disponible en los grupos de usuarios de Amazon Cognito.	Ahora puede enviar solicitudes a las operaciones de API de los grupos de usuarios de Amazon Cognito a través de puntos de enlace de VPC.	4 de noviembre de 2025
Ahora, los grupos de usuarios de Amazon Cognito admiten la vinculación de recursos.	Sus servidores de recursos ahora pueden realizar la verificación de la audiencia de los tokens de acceso con la vinculación de recursos RFC 8707.	24 de octubre de 2025
Se ha añadido un capítulo sobre resolución de problemas .	Nuevo capítulo con soluciones a los problemas más comunes en los grupos de usuarios.	24 de octubre de 2025
Se ha añadido una nueva cuota de solicitudes de documentos con claves web JSON para grupos de usuarios.	Nueva entrada de cuota para la obtención de `.well-known/jwks.json` desde servidores de autorización de OAuth 2.0 para grupos de usuarios.	13 de octubre de 2025

Documentos de términos y condiciones del inicio de sesión administrado.	Ahora puede añadir enlaces a sus términos y condiciones y a su política de privacidad en sus páginas de registro de inicio de sesión administrado.	2 de octubre de 2025
Nuevo ejemplo de aplicación de introducción para grupos de identidades.	Ahora puede configurar una aplicación de ejemplo que muestre las capacidades de los grupos de identidades.	30 de septiembre de 2025
Actualizaciones de los requisitos de las políticas de confianza de roles de IAM para los grupos de identidades.	Ahora, puede realizar cambios en las políticas de confianza de roles de la entidad principal de servicio de grupos de identidades de Amazon Cognito, pero solo cuando una clave de condición aumente el alcance de la audiencia de la federación de OIDC (identidad web) a uno o más grupos de identidades.	1 de agosto de 2025
Amazon Cognito ya está disponible en México (centro). Región de AWS	Ahora puede crear recursos de Amazon Cognito en la región de México (centro).	24 de julio de 2025
Amazon Cognito ya está disponible en Asia Pacífico (Tailandia). Región de AWS	Ahora puede crear recursos de Amazon Cognito en la región de Asia-Pacífico (Tailandia).	24 de julio de 2025

AWS WAF ACL web en el inicio de sesión gestionado.	Ahora puede aplicar reglas de ACL AWS WAF web a los clientes de aplicaciones de grupos de usuarios que tienen la versión de marca de inicio de sesión administrado.	24 de junio de 2025
Se han actualizado los ejemplos de desencadenadores de Lambda.	Se actualizó la función de ejemplo para los activadores Lambda personalizados de correo electrónico y remitente de SMS para que sean compatibles con Node.js la versión 22.x. Ahora, el ejemplo también es más accesible para realizar pruebas.	19 de mayo de 2025
Nuevo parámetro `prompt`.	Ahora tiene un mayor control sobre la reautenticación de las sesiones de inicio de sesión administrado existentes con el parámetro `prompt`. También puede transferir los valores de este parámetro a proveedores externos.	15 de mayo de 2025
Metadatos de cliente para solicitudes M2M.	Ahora puede pasar los metadatos del cliente en las solicitudes de credenciales de cliente o de máquina a máquina (M2M). Amazon Cognito transfiere los metadatos del cliente M2M al desencadenador de Lambda Antes de la generación del token.	29 de abril de 2025

Rotación de tokens de actualización.	Ahora puede obtener nuevos tokens de actualización e invalidar los originales en las solicitudes de actualización de los mismos.	22 de abril de 2025
Amazon Cognito ya está disponible en Asia Pacífico (Malasia). Región de AWS	Ahora puede crear recursos de Amazon Cognito en la región de Asia-Pacífico (Malasia).	7 de marzo de 2025
Personalice los tokens de acceso para las identidades de las máquinas.	El desencadenador de Lambda Antes de la generación del token ahora tiene un evento de la versión tres que modifica las notificaciones de token de acceso y los ámbitos de las concesiones de credenciales de cliente para la autorización de máquina a máquina (M2M).	3 de marzo de 2025
Se ha actualizado la información sobre la política administrada `AmazonCognitoPowerUser` de AWS .	Se agregó una AWS End User Messaging SMS operación en la política AWS administrada para los grupos de usuarios avanzados de Amazon Cognito.	27 de febrero de 2025
Se ha actualizado la descripción de la integración de OpenID Connect (OIDC).	Se agregó un diagrama que ilustra cómo Amazon Cognito se autentica con los proveedores de identidad de OIDC.	25 de febrero de 2025

Se ha agregado información sobre la lógica MFA.	Se agregó un diagrama que ilustra cómo Amazon Cognito aplica la configuración de autenticación multifactor (MFA) de su grupo de usuarios a los usuarios en tiempo de ejecución.	25 de febrero de 2025
Se han añadido prácticas recomendadas de seguridad para los grupos de usuarios de Amazon Cognito.	Se agregó una página sobre cómo proteger los secretos y seguir las prácticas recomendadas de seguridad en la configuración de grupos de usuarios.	25 de febrero de 2025
Actualizaciones de los recursos de introducción para grupos de usuarios.	La experiencia de introducción a los grupos de usuarios de Amazon Cognito incluye un nuevo diseño de consola y opciones de aplicación.	21 de noviembre de 2024
Nuevo modelo de precios con planes de características.	Se actualizó el modelo de facturación para los grupos de usuarios. Ahora, las características de seguridad avanzada son la protección contra amenazas. Los componentes de la licencia de características de seguridad avanzadas están ahora en los planes de características Essentials y Plus.	21 de noviembre de 2024
Nueva característica de inicio de sesión administrado.	Se lanzó el inicio de sesión administrado, una actualización de la interfaz de usuario alojada.	21 de noviembre de 2024

Un nuevo método de autenticación y nuevos flujos de autenticación.	Ahora puede iniciar sesión en los grupos de usuarios de Amazon Cognito con claves de acceso y contraseñas de un solo uso.	21 de noviembre de 2024
Se ha actualizado la información sobre `AmazonCognitoUnAuthedIdentitiesSessionPolicy` .	Se trasladaron AWS Key Management Service las operaciones de la política AWS administrada para reducir el alcance de las identidades no autenticadas de la política integrada a la política administrada. AWS	1 de noviembre de 2024
Se ha agregado el parámetro `login_hint` .	Ahora puede añadir una sugerencia de nombre de usuario a las solicitudes de autorización para la interfaz de usuario alojada, el OIDC y Google. IdPs IdPs	3 de octubre de 2024
Nuevas características de seguridad avanzada para la MFA de correo electrónico.	Ahora puede enviar códigos de autenticación multifactor (MFA) por mensaje de correo electrónico con características de seguridad avanzada.	12 de septiembre de 2024

Nuevo contenido y cambios en páginas.	Se han modificado títulos, se ha eliminado contenido innecesario, se ha agregado introducciones basadas en escenarios, se ha movido la referencia a los puntos de conexión de interfaz de usuario alojada e OIDC de grupos de usuarios a la sección de grupos de usuarios.	9 de septiembre de 2024
Se ha actualizado la información sobre `AmazonCognitoUnAuthedIdentitiesSessionPolicy` .	La política AWS gestionada para reducir el alcance de las identidades no autenticadas en los grupos de identidades ahora permite Amazon Location Service.	9 de agosto de 2024
Nueva prevención de amenazas para una autenticación personalizada con desencadenadores de Lambda y detección de amenazas mejorada.	Ahora puede analizar el inicio de sesión con autenticación personalizada con protección contra amenazas y aplicar respuestas de autenticación adaptables. La protección contra amenazas ahora también analiza el tráfico de inicio de sesión para detectar la distancia geográfica imposible entre intentos.	8 de agosto de 2024

Nuevas características de seguridad avanzada para evitar la reutilización de contraseñas y exportar los registros de actividad de los usuarios.	Ahora puede exportar los registros de actividad de los usuarios y establecer una política de historial de contraseñas con caracterí sticas de seguridad avanzada en los grupos de usuarios de Amazon Cognito.	6 de agosto de 2024
Amazon Cognito ya está disponible en el oeste de Canadá (Calgary) y en Asia Pacífico (Hong Kong). Regiones de AWS	Ahora puede crear recursos de Amazon Cognito en las regiones de Oeste de Canadá (Calgary) y Asia-Pacífico (Hong Kong).	9 de julio de 2024
Se ha mejorado la descripci ón del comportamiento de las aplicaciones para una seguridad avanzada.	Se ha actualizado la informaci ón sobre los datos contextua les del dispositivo para una autenticación flexible de seguridad avanzada.	10 de junio de 2024
Se ha agregado soporte para objetos complejos en el desencadenador de Lambda previo al token	Ahora puede añadir matrices y objetos JSON a las notificac iones de token de ID y de acceso.	30 de mayo de 2024
Se ha actualizado la informaci ón sobre Verified Permissions y Amazon Cognito.	Amazon Verified Permissio ns ahora tiene una integraci ón más directa con Amazon Cognito.	15 de mayo de 2024
Multi-Region Identidades verificadas de Amazon SES.	En algunos casos Regiones de AWS sin Amazon SES, los usuarios de Amazon Cognito agrupan el correo electrónico de equilibrio de carga entre dos regiones remotas.	10 de mayo de 2024

Se ha agregado información sobre la autorización de M2M y los costos de administración.	Aprenda a utilizar las concesiones de credenciales de cliente para casos de uso de máquina a máquina (M2M) con 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 de Asia-Pacífico (Melbourne).	4 de abril de 2024
Se ha agregado una aplicación de ejemplo Android 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 de la introducción, escenarios comunes, prácticas recomendadas para varios usuarios y acceso a 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 Medio Oriente (EAU).	8 de marzo de 2024

Nuevas características de SAML y contenido mejorado.	Ahora puede firmar las solicitudes de SAML, cifrar las respuestas de SAML y configurar el SSO de SAML. IdP-initiated	1 de febrero de 2024
Aumentos de cuota disponibles.	Ahora puede adquirir capacidad adicional para las cuotas de tasa de solicitudes de Amazon Cognito.	25 de enero de 2024
Los grupos de identidades de Amazon Cognito admiten las tasas de solicitudes en Service Quotas.	Ahora puede supervisar las cuotas de solicitudes por segundo (RPS) de los grupos de identidades de Amazon Cognito y aumentar las solicitudes en la consola de Service Quotas.	19 de diciembre de 2023
Se ha agregado una nueva característica 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
Contenido mejorado sobre los clientes de aplicación y los ámbitos de OAuth.	Ediciones de claridad y correcciones a Application-specific configuración con clientes de aplicaciones y Ámbitos, M2M y servidores de recursos. Eliminadas las instrucciones de consola heredadas.	14 de noviembre de 2023

Se ha mejorado el contenido sobre los dispositivos y su autenticación.	Nuevo contenido sobre la utilización de claves de dispositivos y autenticación de la SRP de dispositivos.	18 de octubre de 2023
Guía actualizada. Consola de administración de AWS	Se ha eliminado la referenci a de la consola de grupos de usuarios y redistrib uido 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
De-emphasized acceso directo al punto final LOGIN.	Se ha agregado información general visual del grupo de usuarios Punto de conexión Login y se ha enfatizad o el inicio de la autentica ción 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 ha introducido informaci ón sobre la autorización de Amazon Cognito con Amazon Verified Permissions.	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 la 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 a la página principal del servicio y a la página principal de los grupos de usuarios.	Se han actualizado las páginas de información general 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 identidad de Amazon Cognito en 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
Se han actualizado los ejemplos y las descripci ones de los disparadores de 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 autentica n 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 código de ejemplo para su aplicació n Amazon Cognito en varios lenguajes de programación.	23 de enero de 2023
Se ha agregado información sobre los modelos de API y la autenticación con 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 ha agregado información sobre la protección contra la eliminación de grupos de usuarios.	Al crear un nuevo grupo de usuarios con el Consola de administración de AWS, ahora está protegido contra la eliminación de forma predeterminada.	20 de octubre de 2022
Se ha agregado una guía de usuario para la interfaz de usuario alojada e información sobre la MFA de la 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 visualizar previamente 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 ha agregado 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

Se ha actualizado la documentación de federación. Nueva característica 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 ha agregado nueva información sobre las huellas digitales de los dispositivos para una seguridad avanzada.	31 de mayo de 2022
Inicio de sesión de usuarios federados sin interacción con la interfaz de usuario alojada	Se ha agregado una nueva página sobre cómo marcar aplicaciones, de modo que Amazon Cognito dirija silenciosamente a los usuarios a un inicio de sesión federado.	29 de mayo de 2022
In-Region Mensajes SMS y de correo electrónico 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 en la página de cuotas	Se han agregado y aclarado cuotas de recursos y tasas de solicitudes.	10 de enero de 2022
Nueva experiencia de la consola para 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
prácticas recomendadas de Multi-tenant	Se han agregado prácticas recomendadas para aplicacio nes de varios inquilinos.	4 de marzo de 2021
Atributos para controlar el acceso	Los grupos de identidades de Amazon Cognito proporcio nan 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
Desencadenador de Lambda para remitentes personali zados de SMS y desencade nador de Lambda para remitentes personalizados de correos electrónicos	Con el desencadenador de Lambda para remitente s personalizados de SMS y el desencadenador para remitentes personalizados de correos electrónicos, se puede habilitar a un proveedor de terceros para enviar notificac iones de correo electrónico y SMS a sus usuarios desde el código de función de Lambda.	30 de noviembre de 2020

Actualizaciones de tokens de Amazon Cognito	Se agregó la información actualizada sobre el vencimiento a los tokens de acceso, ID y actualización.	29 de octubre de 2020
Service Quotas de Amazon Cognito	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 las 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 ahora es compatible en EE. UU. AWS GovCloud	Amazon Cognito ahora es compatible con la región AWS GovCloud (EE. UU.).	13 de mayo de 2020

Actualizaciones del documento 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 sobre las medidas de seguridad especiales de 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
Los grupos de identidades de Amazon Cognito ahora admiten Sign in with Apple	Iniciar sesión con Apple está disponible en todas las regiones en las que 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 sobre la sensibilidad a mayúsculas en el nombre de usuario	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 ha agregado información sobre la integración de Amazon Cognito en la aplicación web o móvil mediante los SDK y las bibliotecas de AWS Amplify . 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 desencadenadores de grupos 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 de la ListUsers 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
Compatibilidad del etiquetado	Se agregó información sobre el etiquetado de recursos de Amazon Cognito.	26 de marzo de 2019
Cambios en 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
Se han actualizado los límites	Se han actualizado los límites flexibles de los grupos de usuarios.	11 de diciembre de 2018

Actualización de la documentación sobre la verificación de direcciones de correo electrónico y 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 sobre correos electrónicos de prueba	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

Proveedor de identidades OIDC para grupos de usuarios de Amazon Cognito	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
Desencadenador de migración de Lambda de Amazon Cognito	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 reorganiz ado 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

Seguridad avanzada beta 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 de 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

Características de IU de aplicación integrada y de federación de grupos de usuario 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 de característica relacionados con la conformidad con HIPAA y 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
Características de control de acceso basadas en grupos de usuarios y en roles	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	Se han actualizado los ejemplos de código iOS.	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
Característica 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
Característica 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
Compatibilidad con 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 integration	Se agregó la integración 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
Flujo de datos a Amazon Kinesis	Proporciona control e información de los flujos de datos.	4 de marzo de 2015
Compatibilidad con OpenID Connect	Activa la compatibilidad con los proveedores de OpenID Connect.	23 de noviembre de 2014
Sincronización mediante inserción	Activa la compatibilidad con la sincronización mediante inserción silenciosa.	6 de noviembre de 2014
Developer-authenticated se agregó soporte para identidad es	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.