aws

Guía del usuario

Amazon Verified Permissions

Amazon Verified Permissions: Guía del usuario

Table of Contents

¿Qué es Amazon Verified Permissions?

Amazon Verified Permissions es un servicio de autorización y administración de permisos escalable y detallado para aplicaciones personalizadas diseñado para usted. Verified Permissions permite a sus desarrolladores crear aplicaciones seguras con mayor rapidez al externalizar la autorización y centralizar la gestión y la administración de las políticas. Verified Permissions utiliza el lenguaje de políticas de Cedar para definir permisos detallados para los usuarios de las aplicaciones.

Temas

Autorización en Verified Permissions

Verified Permissions proporciona autorización al verificar si una entidad principal puede realizar una acción en un recurso en un contexto determinado en una aplicación personalizada. Verified Permissions supone que la entidad principal ha sido identificada y autenticada previamente por otros medios, como el uso de protocolos como OpenID Connect, un proveedor hospedado como Amazon Cognito u otra solución de autenticación. Verified Permissions es independiente del lugar donde se administre al usuario y de cómo se ha autenticado.

Verified Permissions es un servicio que permite a los clientes crear, mantener y probar políticas en la AWS Management Console. Los permisos se expresan utilizando el lenguaje de políticas de Cedar. La aplicación cliente solicita la autorización APIs para evaluar las políticas de Cedar almacenadas en el servicio y decidir si se permite una acción o no.

Lenguaje político de Cedar

Las políticas de autorización de Verified Permissions se redactan utilizando el lenguaje de políticas de Cedar. Cedar es un lenguaje de código abierto para redactar políticas de autorización y tomar

decisiones de autorización basadas en esas políticas. Al crear una aplicación, debe asegurarse de que solo los usuarios autorizados puedan acceder a ella y que solo puedan hacer lo que cada usuario está autorizado a hacer. Con Cedar, puede desvincular la lógica empresarial de la lógica de autorización. En el código de la aplicación, preceda las solicitudes realizadas a sus operaciones con una llamada al motor de autorización de Cedar con esta pregunta: "¿Está autorizada esta solicitud?". Después, la aplicación puede realizar la operación solicitada si la decisión es "permitir" o devolver un mensaje de error si la decisión es "denegar".

En la actualidad, Verified Permissions utiliza la versión 2.4 de Cedar.

Para obtener más información sobre Cedar, consulte lo siguiente:

- Guía de referencia sobre el lenguaje de políticas de Cedar
- GitHubRepositorio Cedar

Ventajas de Verified Permissions

Agilizar el desarrollo de las aplicaciones

Agilice el desarrollo de las aplicaciones separando la autorización de la lógica empresarial.

Aplicaciones más seguras

Verified Permissions permite a los desarrolladores crear aplicaciones más seguras.

Características para el usuario final

Verified Permissions permite ofrecer características de usuario final más completas para la administración de permisos.

Servicios relacionados

- 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 AWS credenciales y credenciales de acceso OAuth 2.0. Al crear un almacén de políticas, tiene la opción de crear sus directores y grupos a partir de un grupo de usuarios de Amazon Cognito. Para obtener más información, consulte la Guía para desarrolladores de Amazon Cognito.

- Amazon API Gateway: Amazon API Gateway es un AWS servicio para crear, publicar, mantener, supervisar y proteger RESTHTTP, y WebSocket APIs a cualquier escala. Al crear un almacén de políticas, tiene la opción de crear sus acciones y recursos a partir de un almacén integrado API en API Gateway. Para obtener más información sobre API Gateway, consulte la Guía para desarrolladores de API Gateway.
- AWS IAM Identity Center— Con IAM Identity Center, puede gestionar la seguridad de inicio de sesión de las identidades de sus empleados, también conocidos como usuarios de los empleados. IAMIdentity Center ofrece un lugar en el que puede crear o conectar a los usuarios de la fuerza laboral y gestionar de forma centralizada su acceso a todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte la AWS IAM Identity Center Guía del usuario de .

Acceso a Verified Permissions

Puede trabajar con Amazon Verified Permissions de cualquiera de las siguientes formas:

AWS Management Console

La consola es una interfaz basada en navegador para administrar Verified Permissions y los recursos de AWS . Para obtener más información acerca de cómo acceder a Verified Permissions mediante la consola, consulte Cómo iniciar sesión en AWS en la Guía del usuario de AWS Sign-In .

- Consola de permisos verificados de Amazon

AWS Herramientas de línea de comandos

Puede utilizar las herramientas de línea de AWS comandos para ejecutar comandos en la línea de comandos del sistema a fin de ejecutar AWS tareas y permisos verificados. El uso de la línea de comandos puede ser más rápido y cómodo que la consola. Las herramientas de línea de comandos también son útiles si desea crear scripts que realicen tareas de AWS .

AWS proporciona dos conjuntos de herramientas de línea de comandos: el AWS Command Line Interface(AWS CLI) y el AWS Tools for Windows PowerShell. Para obtener información sobre la instalación y el uso de AWS CLI, consulte la Guía del AWS Command Line Interface usuario. Para obtener información sobre la instalación y el uso de las herramientas para Windows PowerShell, consulte la Guía del AWS Tools for Windows PowerShell usuario.

- permisos verificados en la Referencia de comandos AWS CLI
- Permisos verificados por Amazon en AWS Tools for Windows PowerShell

AWS SDKs

AWS proporciona SDKs (kits de desarrollo de software) que consisten en bibliotecas y código de muestra para varios lenguajes de programación y plataformas (Java, Python, Ruby,. NET, iOS, Android, etc.). SDKsProporcionan una forma cómoda de crear un acceso programático a los permisos verificados y AWS. Por ejemplo, se SDKs encargan de tareas como firmar criptográficamente las solicitudes, gestionar los errores y volver a intentar las solicitudes automáticamente.

Para obtener más información y descargarla AWS SDKs, consulte Herramientas para. Amazon Web Services

Los siguientes son enlaces a la documentación de varios recursos sobre permisos verificados AWS SDKs.

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto)
- AWS SDK for Ruby

Constructos AWS CDK

AWS Cloud Development Kit (AWS CDK) Se trata de un marco de desarrollo de software de código abierto para definir la infraestructura de nube en el código y aprovisionarla mediante ella. AWS CloudFormation Se pueden usar construcciones, o componentes de nube reutilizables, para crear plantillas. AWS CloudFormation Luego, estas plantillas se pueden usar para implementar su infraestructura de nube.

Para obtener más información y descargar AWS CDKs, consulte AWS Cloud Development Kit.

Los siguientes son enlaces a la documentación de los AWS CDK recursos de permisos verificados, como las construcciones.

- Amazon Verified Permissions L2 Construct CDK

Permisos verificados API

Puede acceder a los permisos verificados y mediante AWS programación mediante los permisos verificadosAPI, que le permiten emitir HTTPS solicitudes directamente al servicio. Cuando utilices elAPI, debes incluir un código para firmar digitalmente las solicitudes con tus credenciales.

- Guía de API referencia de permisos verificados de Amazon

Precios de Verified Permissions

Verified Permissions ofrece precios escalonados en función de la cantidad de solicitudes de autorización que realicen al mes sus solicitudes a Verified Permissions. También hay precios para las acciones de gestión de políticas que se basan en la cantidad de API solicitudes de políticas c URL (de clientesURL) al mes que realizan tus solicitudes a permisos verificados.

Para obtener una lista completa de los cargos y precios de Verified Permissions, consulte Precios de Amazon Verified Permissions.

Para ver su factura, vaya al Panel de Billing and Cost Management en la consola de AWS Billing and Cost Management. La factura contiene vínculos a informes de uso que ofrecen detalles sobre la cuenta. Para obtener más información sobre la Cuenta de AWS facturación, consulta la Guía AWS Billing del usuario.

Si tiene preguntas sobre la AWS facturación, las cuentas y los eventos, póngase en contacto con AWS Support.

Cómo empezar con los permisos verificados de Amazon

Para empezar con los permisos verificados, necesita una AWS cuenta y un centro de IAM identidad con usuarios con los permisos necesarios para crear recursos en los permisos verificados.

Las siguientes secciones le ayudarán a crear una AWS cuenta y los usuarios necesarios:

Inscríbase en una Cuenta de AWS

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

Para suscribirte a una Cuenta de AWS

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

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

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

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

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

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

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Activa la autenticación multifactorial (MFA) para tu usuario root.

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

Creación de un usuario con acceso administrativo

1. Habilite IAM Identity Center.

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

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

Iniciar sesión como usuario con acceso de administrador

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

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

Concesión de acceso a usuarios adicionales

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

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

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

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

Ahora que ha creado una AWS cuenta y algunos usuarios, está listo para crear un almacén de políticas. Elige una de las siguientes opciones para empezar a utilizar los permisos verificados:

- Crea tu primera tienda de políticas de permisos verificados de Amazon
- Crear un almacén de políticas para usar API Gateway con un proveedor de identidades

Crea tu primera tienda de políticas de permisos verificados de Amazon

Al iniciar sesión en la consola de Verified Permissions por primera vez, puede elegir cómo crear su primer almacén de políticas y la política de Cedar. Siga el procedimiento de inicio de sesión apropiado para su tipo de usuario como se describe en el tema Cómo iniciar sesión en AWS en la Guía del usuario de inicio de sesión en AWS . En la página de inicio de la consola, seleccione el servicio Amazon Verified Permissions. Elija Comenzar.

Creación de un almacén de políticas de muestra

Si es la primera vez que utiliza Verified Permissions, le recomendamos que utilice uno de los almacenes de políticas de muestra para familiarizarse con el funcionamiento de Verified Permissions. Los almacenes de políticas de muestra proporcionan políticas predefinidas y un esquema.

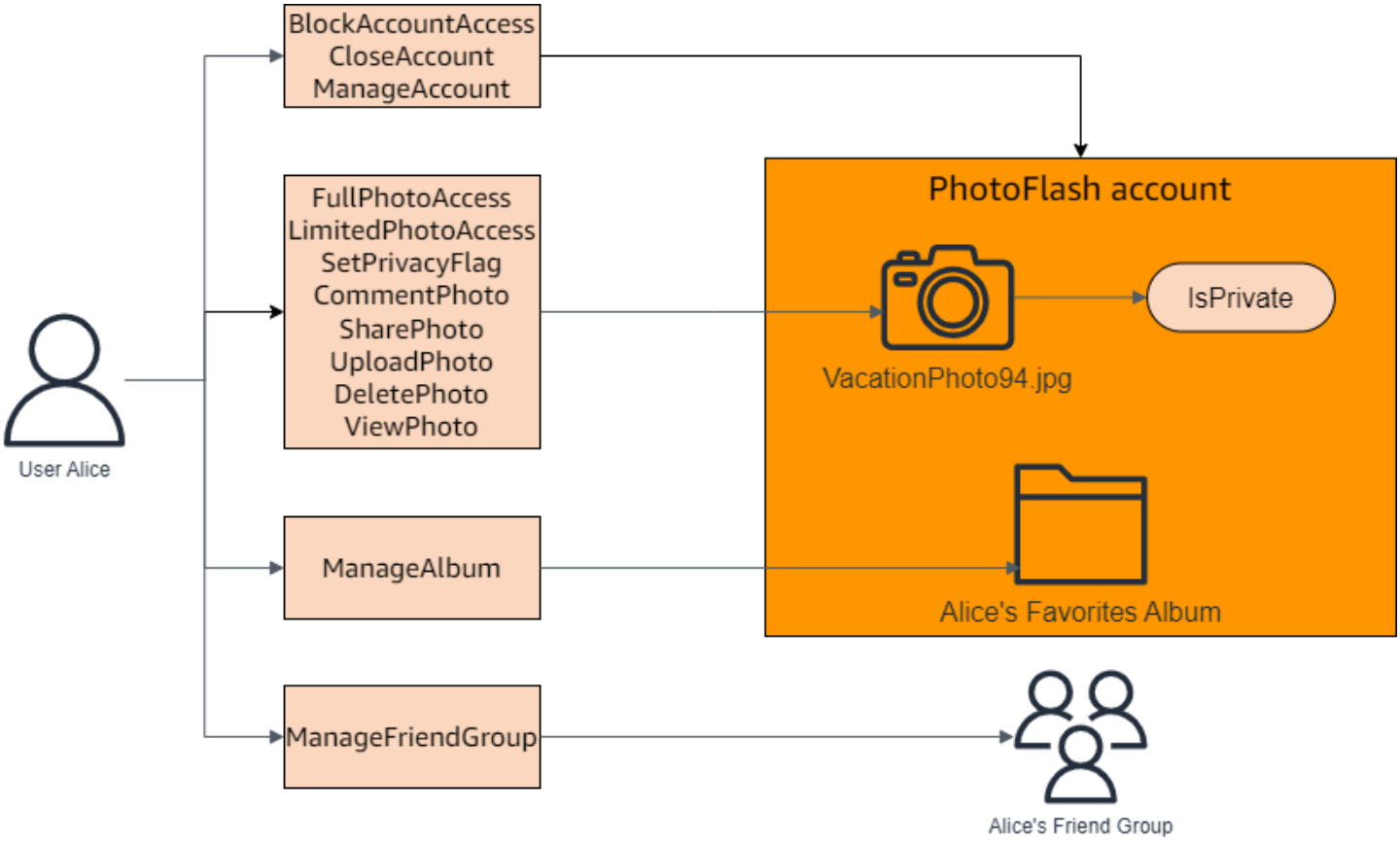
Para crear un almacén de políticas con el método de configuración Almacén de políticas de muestra

1. En la consola de permisos verificados, seleccione Crear un nuevo almacén de políticas.
2. En la sección Opciones de inicio, elija un almacén de políticas de muestra.
3. En la sección Ejemplo de proyecto, elija el tipo de aplicación de Verified Permissions de muestra que va a utilizar. Para este tutorial, elija el almacén PhotoFlashde políticas.

4. Se generará automáticamente un espacio de nombres para el esquema del almacén de políticas de muestra en función del proyecto de ejemplo que haya elegido.
5. Seleccione Crear almacén de políticas.

   El almacén de políticas se crea con políticas, plantillas de políticas y un esquema para el almacén de políticas de muestra.

El siguiente diagrama ilustra las relaciones entre las acciones del almacén de políticas de PhotoFlash muestra y los tipos de recursos a los que se aplican.



Creación de políticas vinculadas a plantillas para un almacén de políticas de muestra

El almacén de políticas de PhotoFlash muestra incluye políticas, plantillas de políticas y un esquema. Puede crear políticas vinculadas a plantillas a partir de las plantillas de políticas incluidas en el almacén de políticas de muestra.

Para crear políticas vinculadas a plantillas para el almacén de políticas de muestra

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Políticas.
3. Seleccione Crear política y, a continuación, elija Crear política vinculada a una plantilla.
4. Selecciona el botón de radio situado junto a la plantilla de política con la descripción Otorgar acceso total a las fotos compartidas no privadas y, a continuación, selecciona Siguiente.
5. Para Principal, ingresa`PhotoFlash::User::"Alice"`. Para Recurso, introduzca`PhotoFlash::Album::"Bob-Vacation-Album"`.
6. Seleccione Crear política vinculada a una plantilla.

   La nueva política vinculada a una plantilla se muestra en Políticas.
7. Cree otra política vinculada a una plantilla para el almacén de políticas de PhotoFlash muestra. Seleccione Crear política y, a continuación, elija Crear política vinculada a una plantilla.
8. Selecciona el botón de opción situado junto a la plantilla de política con la descripción Otorgar acceso limitado a las fotos compartidas no privadas y, a continuación, selecciona Siguiente.
9. Para Principal, ingresa`PhotoFlash::FriendGroup::"MySchoolFriends"`. Para Recurso, introduzca`PhotoFlash::Album::"Alice's favorite album"`.
10. Seleccione Crear política vinculada a una plantilla.

    La nueva política vinculada a una plantilla se muestra en Políticas.

Probaremos las nuevas políticas vinculadas a plantillas en la siguiente sección del tutorial. Para ver más ejemplos de valores para los que puede crear una política vinculada a una plantilla, consulte. PhotoFlash PhotoFlashejemplos

Prueba de un almacén de políticas de muestra

Tras crear el almacén de políticas de muestra y las políticas vinculadas a plantillas, puede probar las políticas estáticas de muestra de Verified Permissions y las nuevas políticas vinculadas a plantillas ejecutando una simulación de solicitud de autorización mediante el banco de pruebas de Verified Permissions.

En función de cuándo haya creado el almacén de políticas de muestra, las plantillas de políticas pueden diferir de las referencias de este procedimiento. Antes de comenzar esta parte del tutorial,

compruebe que tiene todas las plantillas de políticas siguientes en el almacén de políticas de PhotoFlash ejemplo. Si tu política no se alinea con estas políticas, edita las políticas existentes o crea un nuevo almacén de políticas a partir de la opción Ejemplo de proyecto PhotoFlash.

Concede acceso completo a las fotos compartidas no privadas

```
permit (
    principal in ?principal,
    action in PhotoFlash::Action::"FullPhotoAccess",
    resource in ?resource
)
when { resource.IsPrivate == false };
```

Otorga acceso limitado a las fotos compartidas no privadas

```
permit (
    principal in ?principal,
    action in PhotoFlash::Action::"LimitedPhotoAccess",
    resource in ?resource
)
when { resource.IsPrivate == false };
```

Para probar políticas de muestra del almacén de políticas

1. Abre la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, seleccione Banco de pruebas.
3. Elija Modo visual.
4. En la sección Principal, elige PhotoFlash: :User entre los tipos principales de tu esquema. Escriba un identificador para el usuario en el cuadro de texto. Por ejemplo, `Alice`.
5. No seleccione Agregar un elemento principal para la entidad principal.
6. Para el atributo Account: Entity, asegúrese de que la entidad PhotoFlash: :Account esté seleccionada. Escriba un identificador para la cuenta. Por ejemplo, `Alice-account`.
7. En la sección Recursos, elige el tipo de recurso PhotoFlash: :Photo. Escriba un identificador para la foto en el cuadro de texto. Por ejemplo, `photo.jpeg`.
8. Elija Añadir un elemento principal y elija PhotoFlash: :Cuenta para el tipo de entidad. Escriba el mismo identificador para la cuenta principal de la foto que ha especificado en el campo Account: Entity para el usuario. Por ejemplo, `Alice-account`.

9. En la sección Acción, selecciona PhotoFlash: :Action::»ViewPhoto» de la lista de acciones válidas.
10. En la sección Entidades adicionales, seleccione Agregar esta entidad para añadir la entidad de cuenta sugerida.
11. Seleccione Ejecutar solicitud de autorización en la parte superior de la página para simular la solicitud de autorización para las políticas de Cedar en el almacén de políticas de muestra. El banco de pruebas debe mostrar la decisión para permitir la solicitud.

La siguiente tabla proporciona valores adicionales para la entidad principal, el recurso y la acción que puede probar con el banco de pruebas de Verified Permissions. La tabla incluye la decisión de solicitud de autorización basada en las políticas estáticas incluidas en el almacén de políticas de PhotoFlash muestra y en las políticas vinculadas a plantillas que creó en la sección anterior.

Valor de entidad principal	Valor Account: Entity de entidad principal	Valor de recurso	Valor de elemento principal del recurso	Action	Decisión de autorización
PhotoFlash: :User | Alice	PhotoFlash: :Cuenta | Cuenta Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Cuenta | Cuenta BOB	PhotoFlash: :Acción::»» ViewPhoto	Denegar
PhotoFlash: :Usuario | Alice	PhotoFlash: :Cuenta | Cuenta Alice	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Cuenta | Cuenta Alice	PhotoFlash: :Acción::»» ViewPhoto	Permitir
PhotoFlash: :Usuario | Alice	PhotoFlash: :Cuenta | Cuenta Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash:: Álbum | Bob-Vacation-Album	PhotoFlash: :Acción::»» ViewPhoto	Permitir
PhotoFlash: :Usuario | Alice	PhotoFlash: :Cuenta | Cuenta Alice	PhotoFlash: :Foto | Bob-photo .jpeg	PhotoFlash:: Álbum | Bob-Vacation-Album	PhotoFlash: :Acción::»» DeletePhoto	Denegar

Valor de entidad principal	Valor Account: Entity de entidad principal	Valor de recurso	Valor de elemento principal del recurso	Action	Decisión de autorización
PhotoFlash: :Usuario | Alice	PhotoFlash: :Cuenta | Cuenta Alice	PhotoFlash: :Photo | Bob-photo.jpeg,IsPrivate: Booleano | verdadero	PhotoFlash: :Álbum | Bob-Vacation-Album	PhotoFlash: :Acción::»» ViewPhoto	Denegar
PhotoFlash: :Usuario | Jane, PhotoFlash:: | FriendGroup MySchoolFriends	PhotoFlash: :Cuenta | Cuenta Jane	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | El álbum favorito de Alice	PhotoFlash: :Action::»» ViewPhoto	Permitir
PhotoFlash: :Usuario | Jane, PhotoFlash:: | FriendGroup MySchoolFriends	PhotoFlash: :Cuenta | Cuenta Jane	PhotoFlash: :Foto | photo.jpeg	PhotoFlash: :Album | El álbum favorito de Alice	PhotoFlash: :Action::»» DeletePhoto	Denegar

Crear un almacén de políticas para usar API Gateway con un proveedor de identidades

Un caso de uso común es utilizar los permisos verificados de Amazon para autorizar el acceso de los usuarios a los APIs alojamientos alojados en Amazon API Gateway. Con un asistente en la AWS consola, puede crear políticas de acceso basadas en roles para los usuarios gestionados en

Amazon Cognito o en OIDC cualquier proveedor de identidad (IdP), e implementar AWS Lambda un autorizador que llame a Verified Permissions para evaluar estas políticas.

Para completar el asistente, elija Configurar con API Gateway y un proveedor de identidad al crear un nuevo almacén de políticas y siga los pasos.

Se crea un almacén de políticas API vinculado que aprovisiona el modelo de autorización y los recursos para las solicitudes de autorización. El almacén de políticas tiene una fuente de identidad y un autorizador Lambda que conecta API Gateway a Verified Permissions. Una vez creado el almacén de políticas, puede autorizar API las solicitudes en función de la pertenencia a grupos de usuarios. Por ejemplo, los permisos verificados solo pueden conceder acceso a los usuarios que son miembros del `Directors` grupo.

A medida que su aplicación crezca, podrá implementar una autorización detallada con atributos de usuario y ámbitos OAuth 2.0 utilizando el lenguaje de políticas de Cedar. Por ejemplo, los permisos verificados solo pueden conceder acceso a los usuarios que tienen un `email` atributo en el dominio. `mycompany.co.uk`

Una vez que haya configurado su modelo de autorizaciónAPI, su responsabilidad restante será autenticar a los usuarios y generar API solicitudes en su aplicación, así como mantener su almacén de políticas.

Para obtener más información, consulte API-tiendas de pólizas vinculadas.

Mejores prácticas para diseñar un modelo de autorización

Mientras realiza los preparativos para utilizar el servicio Amazon Verified Permissions en una aplicación de software, puede resultar difícil pasar inmediatamente a redactar instrucciones de política como primer paso. Esto sería similar a comenzar el desarrollo de otras partes de una aplicación escribiendo SQL declaraciones o API especificaciones antes de decidir por completo lo que debe hacer la aplicación. En su lugar, debería empezar con una experiencia de usuario para tener una idea clara de lo que los usuarios finales deberían ver cuando gestionen los permisos en la interfaz de usuario de la aplicación. Después, partiendo de esa experiencia vaya hacia atrás para desarrollar una estrategia de implementación.

A medida que vaya realizando este trabajo, se hará preguntas como las siguientes:

- ¿Cuáles son mis recursos? ¿Tienen relaciones entre sí? Por ejemplo, ¿los archivos se encuentran dentro de una carpeta?
- ¿Qué acciones pueden realizar las entidades principales en cada recurso?
- ¿Cómo adquieren esos permisos las entidades principales?
- ¿Desea que sus usuarios finales elijan entre permisos predefinidos, como «Administrador», «Operador» u «ReadOnly», o deberían crear declaraciones de políticas ad hoc? ¿O ambos?
- ¿Los permisos deben heredarse entre recursos, como los archivos que heredan los permisos de una carpeta principal?
- ¿Qué tipos de consultas son necesarios para ofrecer la experiencia de usuario? Por ejemplo, ¿necesita enumerar todos los recursos a los que puede acceder una entidad principal para mostrar la página de inicio de ese usuario?
- ¿Pueden los usuarios quedarse sin acceso a sus propios recursos por accidente? ¿Es necesario evitarlo?

El resultado final de este ejercicio se denomina modelo de autorización; define las entidades principales, los recursos, las acciones y la forma en que se interrelacionan entre sí. La elaboración de este modelo no requiere un conocimiento exclusivo de Cedar o del servicio Verified Permissions. Por el contrario, se trata ante todo de un ejercicio de diseño de la experiencia de usuario, como cualquier otro, y puede manifestarse en artefactos como maquetas de interfaces, diagramas lógicos y una descripción general de cómo los permisos influyen en lo que los usuarios ven en el producto. Cedar está diseñado para ser lo suficientemente flexible como para satisfacer las necesidades de los clientes en un modelo, en lugar de forzar al modelo a flexibilizarse de forma poco natural

para adaptarse a una implementación de Cedar. Por eso, tener una idea clara de la experiencia de usuario que se desea es la mejor manera de llegar a un modelo óptimo.

En esta sección se proporcionan instrucciones generales sobre cómo abordar el ejercicio de diseño, los aspectos a tener en cuenta y una recopilación de las mejores prácticas para utilizar correctamente Verified Permissions.

Además de las pautas que se presentan aquí, tenga siempre en cuenta las mejores prácticas incluidas en la guía de referencia lingüística sobre las políticas de Cedar.

Temas

No existe un modelo canónico “correcto”

Al diseñar un modelo de autorización, no hay una respuesta única y única que sea correcta. Diferentes aplicaciones pueden utilizar de manera efectiva diferentes modelos de autorización para conceptos similares, y es algo que está bien. Por ejemplo, fíjese en la representación del sistema de archivos de un ordenador. Al crear un archivo en un sistema operativo similar a Unix, no hereda automáticamente los permisos de la carpeta principal. Por el contrario, en otros muchos sistemas operativos y en la mayoría de los servicios de intercambio de archivos en línea, los archivos heredan los permisos de su carpeta principal. Ambas opciones son válidas en función de las circunstancias para las que se esté optimizando la aplicación.

La corrección de una solución de autorización no es absoluta, pero debe considerarse en términos de cómo ofrece la experiencia que desean sus clientes y de si protege sus recursos de la manera que esperan. Si su modelo de autorización cumple con estos requisitos, entonces es correcto.

Por eso, empezar el diseño con la experiencia de usuario deseada es el requisito previo más útil para crear un modelo de autorización eficaz.

Céntrese en sus recursos más allá de las operaciones API

En la mayoría de las aplicaciones orientadas al consumidor, los permisos se modelan en función de los recursos compatibles con la aplicación. Por ejemplo, una aplicación para compartir archivos puede representar los permisos como acciones que se pueden realizar en un archivo o una carpeta. Se trata de un modelo bueno y sencillo que abstrae la implementación subyacente y las operaciones de backendAPI.

Por el contrario, otros tipos de aplicaciones, especialmente los servicios web, suelen diseñar los permisos en función de API las propias operaciones. Por ejemplo, si un servicio web proporciona un API nombre`createThing()`, el modelo de autorización puede definir el permiso correspondiente o un nombre `action` en Cedar`createThing`. Esto funciona en muchas situaciones y hace que sea más fácil entender los permisos. Para invocar la operación `createThing`, necesita el permiso de acción `createThing`. Parece sencillo, ¿verdad?

Descubrirá que el proceso de inicio de la consola de permisos verificados incluye la opción de crear sus recursos y acciones directamente a partir de unAPI. Se trata de una base útil: un mapeo directo entre el almacén de políticas y el almacén al API que autoriza.

Sin embargo, este enfoque API centrado puede no ser óptimo, ya APIs que no es más que un indicador de lo que sus clientes realmente están intentando proteger: los datos y los recursos subyacentes. Si varios APIs controlan el acceso a los mismos recursos, puede resultar difícil para los administradores razonar sobre las rutas hacia esos recursos y gestionar el acceso en consecuencia.

Por ejemplo, pensemos en un directorio de usuarios que contenga los miembros de una organización. Los usuarios se pueden organizar en grupos y uno de los objetivos de seguridad es impedir que personas no autorizadas descubran la pertenencia a esos grupos. El servicio que administra este directorio de usuarios proporciona dos API operaciones:

- `listMembersOfGroup`
- `listGroupMembershipsForUser`

Los clientes pueden usar cualquiera de estas operaciones para descubrir la pertenencia a un grupo. Por lo tanto, el administrador de los permisos debe acordarse de coordinar el acceso a ambas operaciones. Esto se complica aún más si más adelante decide agregar una nueva API operación para abordar casos de uso adicionales, como los siguientes.

- `isUserInGroups`(una nueva API herramienta para comprobar rápidamente si un usuario pertenece a uno o más grupos)

Desde el punto de vista de la seguridad, esto API abre una tercera vía para descubrir las pertenencias a grupos, lo que interrumpe los permisos cuidadosamente diseñados del administrador.

Le recomendamos que ignore la API semántica y, en cambio, se centre en los datos y recursos subyacentes y en sus operaciones de asociación. Al aplicar este enfoque al ejemplo de pertenencia a un grupo, se obtendría un permiso abstracto`viewGroupMembership`, como el que debe consultar cada una de las tres API operaciones.

APINombre	Permisos
`listMembersOfGroup`	requiere el permiso `viewGroupMembership` del grupo
`listGroupMembershipsForUser`	requiere el permiso `viewGroupMembership` del usuario
`isUserInGroups`	requiere el permiso `viewGroupMembership` del usuario

Al definir este permiso, el administrador puede controlar el acceso al descubrimiento de la pertenencia a un grupo, ahora y siempre. Como compensación, cada API operación ahora debe documentar los posibles permisos que necesite, y el administrador debe consultar esta documentación al crear los permisos. Pero puede ser una desventaja válida cuando sea necesaria para cumplir tus requisitos de seguridad.

La autorización compuesta es normal

La autorización compuesta se produce cuando la actividad de un solo usuario, como hacer clic en un botón de la interfaz de la aplicación, requiere varias consultas de autorización individuales para determinar si esa actividad está permitida. Por ejemplo, mover un archivo a un nuevo directorio de un sistema de archivos puede requerir tres permisos diferentes: la capacidad de eliminar un archivo

del directorio de origen, la capacidad de añadir un archivo al directorio de destino y, posiblemente, la capacidad de manipular el archivo en sí (según la aplicación).

Si es la primera vez que diseña un modelo de autorización, podría pensar que todas las decisiones de autorización deben poder resolverse en una única consulta de autorización. Sin embargo, esto puede llevar a modelos demasiado complejos y a instrucciones de políticas complicadas. En la práctica, el uso de autorizaciones compuestas puede resultar útil para crear un modelo de autorización más simple. Un indicador de un modelo de autorización bien diseñado es que, cuando se han descompuesto suficientemente las acciones individuales, las operaciones compuestas, como mover un archivo, se pueden representar mediante una agregación intuitiva de primitivas.

Otra situación en la que se produce una autorización compuesta es cuando varias partes participan en el proceso de concesión de un permiso. Considere la posibilidad de crear un directorio organizativo en el que los usuarios puedan ser miembros de grupos. Un enfoque sencillo consiste en dar permiso al propietario del grupo para añadir a cualquier persona. Sin embargo, ¿qué sucede si quiere que los usuarios den antes su consentimiento para que los agreguen? Esto introduce un acuerdo recíproco en el que tanto el usuario como el grupo deben dar su consentimiento para pertenecer a él. Para ello, puede introducir otro permiso vinculado al usuario y especificar si el usuario se puede añadir a cualquier grupo o a un grupo concreto. Cuando un solicitante intente añadir miembros a un grupo después, la aplicación deberá verificar que se cumplen los dos lados del permiso: que la persona solicitante tiene permiso para añadir miembros al grupo especificado y que el usuario individual que se va a añadir tiene los permisos necesarios para añadirlo. Cuando existen acuerdos de `N` partes, es habitual utilizar `N` consultas de autorización compuestas para hacer cumplir cada parte del acuerdo.

Si se enfrenta a un problema de diseño en el que intervienen varios recursos y no tiene claro cómo modelar los permisos, puede ser una señal de que se trata de un escenario de autorización compuesto. En este caso, podría encontrar una solución dividiendo la operación en varias comprobaciones de autorización individuales.

Consideraciones sobre la multitenencia

Es posible que desee desarrollar aplicaciones para que las utilicen varios clientes (empresas que consumen su aplicación o inquilinos) e integrarlas con Amazon Verified Permissions. Antes de desarrollar su modelo de autorización, desarrolle una estrategia multiusuario. Puede administrar las políticas de sus clientes en un almacén de políticas compartido o asignar a cada uno un almacén de políticas por inquilino.

1. Un almacén de políticas compartido

   Todos los inquilinos comparten un único almacén de pólizas. La aplicación envía todas las solicitudes de autorización al almacén de políticas compartido.

2. Almacén de políticas por inquilino

   Cada inquilino tiene un almacén de pólizas dedicado. La aplicación consultará diferentes almacenes de pólizas para tomar una decisión de autorización, en función del inquilino que presente la solicitud.

Ninguna de estas estrategias crea un volumen relativamente mayor de solicitudes de autorización que podrían repercutir en su factura. AWS Entonces, ¿cómo debería diseñar su enfoque? Las siguientes son condiciones comunes que podrían contribuir a su estrategia de autorización de arrendamiento múltiple con permisos verificados.

Políticas de inquilinos: aislamiento

El aislamiento de las políticas de cada inquilino de las demás es importante para proteger los datos del inquilino. Cuando cada inquilino tiene su propio almacén de políticas, cada uno tiene su propio conjunto aislado de políticas.

Flujo de autorización

Puede identificar a un inquilino que realiza una solicitud de autorización con un ID de almacén de políticas en la solicitud, si utiliza almacenes de políticas por inquilino. Con un almacén de políticas compartido, todas las solicitudes utilizan el mismo ID de almacén de políticas.

Administración de plantillas y esquemas

Las plantillas de políticas y el esquema de un almacén de políticas añaden un nivel de sobrecarga de diseño y mantenimiento a cada almacén de políticas.

Administración de políticas globales

Es posible que desee aplicar algunas políticas globales a todos los inquilinos. El nivel de gastos generales de administración de las políticas globales varía según el modelo de almacén de políticas compartido y el modelo por inquilino.

Inquilino abandona el embarque

Algunos inquilinos aportarán elementos a su esquema y políticas que sean específicos para su caso. Cuando un inquilino ya no está activo en su organización y usted desea eliminar sus datos, el nivel de esfuerzo varía en función de su nivel de aislamiento respecto a los demás inquilinos.

Cuotas de recursos de servicio

Verified Permissions tiene cuotas de recursos y tasas de solicitudes que pueden influir en su decisión de tener varios arrendatarios. Para obtener más información sobre las cuotas, consulte Cuotas de recursos.

Comparación de los almacenes de políticas compartidos y los almacenes de políticas por inquilino

Cada consideración requiere su propio nivel de dedicación de tiempo y recursos en los modelos de almacenes de políticas compartidos y por inquilino.

Consideración	Nivel de esfuerzo en un almacén de políticas compartido	Nivel de esfuerzo en los almacenes de pólizas por inquilino
Aislamiento de políticas de inquilinos	Medio.Debe incluir los identificadores de los inquilinos en las políticas y solicitudes de autorización.	Bajo. El aislamiento es el comportamiento predeterminado. Los demás inquilinos no pueden acceder a las políticas específicas para inquilinos.
Flujo de autorización	Bajo. Todas las consultas se dirigen a un almacén de políticas.	Medio. Debe mantener los mapeos entre cada inquilino y su ID de almacén de pólizas.
Administración de plantillas y esquemas	Bajo. Debe hacer que un esquema funcione para todos los inquilinos.	Alto. Los esquemas y las plantillas pueden ser menos complejos individualmente, pero los cambios requieren

		más coordinación y complejidad.
Administración de políticas globales	Bajo. Todas las políticas son globales y se pueden actualizar de forma centralizada.	Alto. Debes añadir políticas globales a cada almacén de políticas durante la incorporación. Replica las actualizaciones de las políticas globales entre muchos almacenes de pólizas.
Inquilino abandona el embarque	Medio. Debe identificar y eliminar únicamente las políticas específicas del inquilino.	Bajo. Elimine el almacén de políticas.
Cuotas de recursos de servicio	Altas. Los inquilinos comparten las cuotas de recursos que afectan a los almacenes de políticas, como el tamaño del esquema, el tamaño de las políticas por recurso y las fuentes de identidad por almacén de políticas.	Bajo. Cada inquilino tiene cuotas de recursos específicas.

¿Cómo elegir

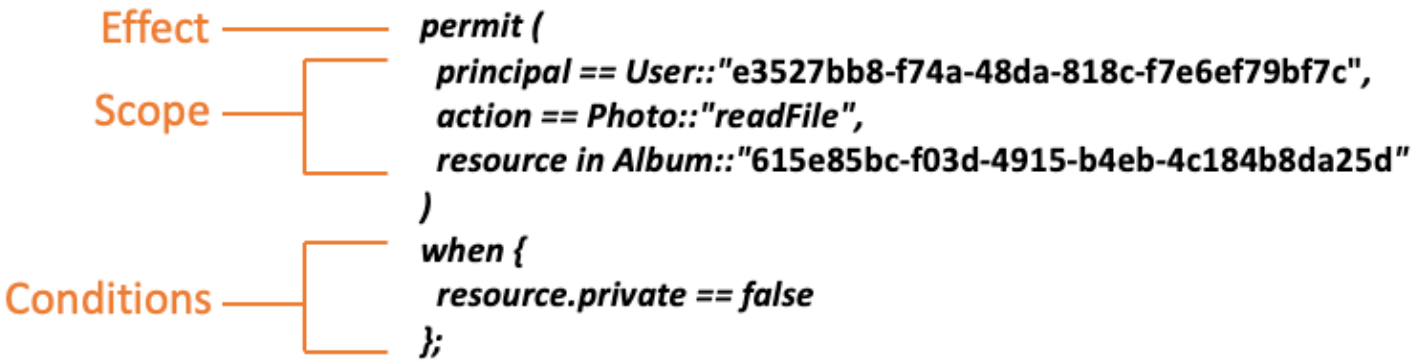
Cada aplicación multiusuario es diferente. Compare cuidadosamente los dos enfoques y sus consideraciones antes de tomar una decisión arquitectónica.

Si su aplicación no requiere políticas específicas para cada inquilino y utiliza una única fuente de identidad, es probable que la solución más eficaz sea un almacén de políticas compartido para todos los inquilinos. Esto se traduce en un flujo de autorización y una gestión de políticas globales más sencillos. La exclusión de un inquilino mediante un almacén de políticas compartido requiere menos esfuerzo, ya que la aplicación no necesita eliminar las políticas específicas del inquilino.

Sin embargo, si su solicitud requiere muchas políticas específicas para cada inquilino o utiliza varias fuentes de identidad, lo más probable es que los almacenes de pólizas por inquilino sean los más eficaces. Puede controlar el acceso a las políticas de inquilinos con IAM políticas que concedan permisos por inquilino a cada almacén de políticas. La exclusión de un inquilino implica eliminar su almacén de pólizas; en un shared-policy-store entorno, debe buscar y eliminar las políticas específicas del inquilino.

Completar, cuando sea posible, el ámbito de la política

El ámbito de la política es la parte de una instrucción de política de Cedar que aparece después de las palabras clave `permit` o `forbid` y entre los paréntesis iniciales.



Le recomendamos que complete los valores de `principal` y `resource` siempre que sea posible. Esto permite a Verified Permissions indexar las políticas para una recuperación más eficiente y, por lo tanto, mejora el rendimiento. Si necesita conceder los mismos permisos a muchas entidades principales o recursos diferentes, le recomendamos que utilice una plantilla de políticas y la adjunte a cada par de entidad principal y recurso.

Evite crear una política amplia que contenga listas de entidades principales y recursos en una cláusula `when`. Si lo hace, es probable que se enfrente a límites de escalabilidad o a problemas operativos. Por ejemplo, para añadir o eliminar un solo usuario de una lista grande de una política, es necesario leer toda la política, editar la lista, redactar la nueva política en su totalidad y gestionar los errores de simultaneidad si un administrador sobrescribe los cambios de otro. Por el contrario, si se utilizan muchos permisos específicos, añadir o eliminar un usuario es tan sencillo como añadir o eliminar la única política que se le aplica.

Todos los recursos deben residir en un contenedor

Al diseñar un modelo de autorización, cada acción debe estar asociada a un recurso concreto. Con una acción como `viewFile`, el recurso al que se puede aplicar es intuitivo: un archivo individual o, quizás, una colección de archivos dentro de una carpeta. Sin embargo, una operación como `createFile` es menos intuitiva. Al modelar la capacidad de crear un archivo, ¿a qué recurso se aplica? No puede ser al archivo en sí, porque el archivo aún no existe.

Este es un ejemplo de los problemas generales que plantea la creación de recursos. La creación de recursos es un problema que surge al inicio. Debe haber una forma de que algo tenga permiso para crear recursos incluso cuando aún no existan recursos. La solución consiste en reconocer que todos los recursos deben existir dentro de algún contenedor, y es el propio contenedor el que actúa como punto de anclaje de los permisos. Por ejemplo, si ya existe una carpeta en el sistema, la capacidad de crear un archivo se puede modelar como un permiso en esa carpeta, ya que es la ubicación en la que se necesitan permisos para crear una instancia del nuevo recurso.

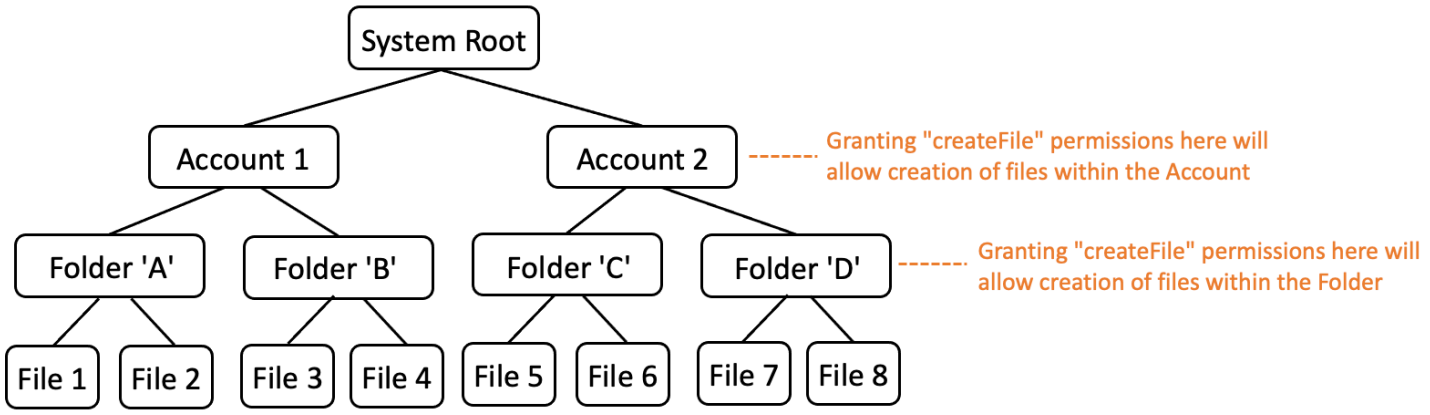
```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource == Folder::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

Pero, ¿qué pasa si no existe ninguna carpeta? Quizás se trate de una cuenta de cliente completamente nueva en una aplicación para la que aún no existen recursos. En esta situación, todavía hay un contexto que se puede entender intuitivamente preguntándose: ¿dónde puede el cliente crear nuevos archivos? No querrá que puedan crear archivos dentro de una cuenta de cliente aleatoria. Más bien, hay un contexto implícito: el límite de la propia cuenta del cliente. Por lo tanto, la cuenta en sí misma representa el contenedor para la creación de recursos, y esto se puede modelar explícitamente en una política similar a la del siguiente ejemplo.

```
// Grants permission to create files within an account,
// or within any sub-folder inside the account.
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    action == Action::"createFile",
    resource in Account::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

Sin embargo, ¿qué pasa si tampoco existen cuentas? Puede optar por diseñar el flujo de trabajo de registro de clientes para que cree nuevas cuentas en el sistema. Si es así, necesitará un contenedor que incluya el límite máximo en el que el proceso puede crear las cuentas. Este contenedor de nivel raíz representa el sistema en su conjunto y podría denominarse algo así como “raíz del sistema”. Sin embargo, la decisión de si es necesario y qué nombre asignarle le corresponde a usted, el propietario de la aplicación.

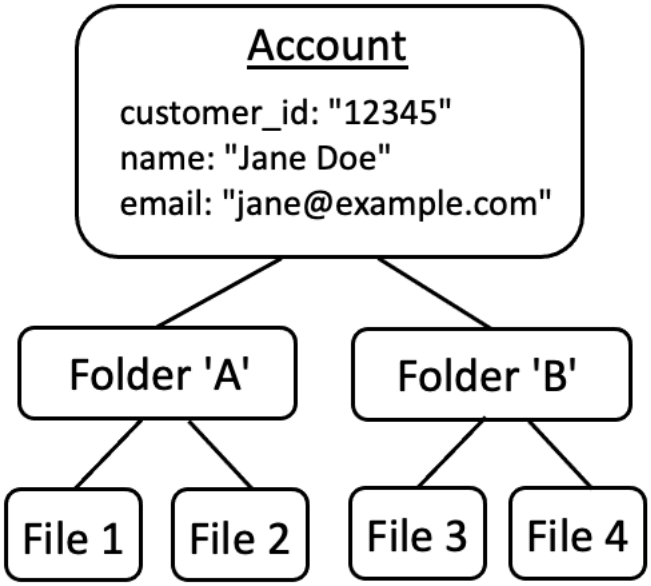
Por lo tanto, en este ejemplo de aplicación, la jerarquía de contenedores resultante sería la siguiente:



Este es un ejemplo de jerarquía. Otras también son válidas. Lo que hay que recordar es que la creación de recursos siempre ocurre en el contexto de un contenedor de recursos. Estos contenedores pueden estar implícitos, como el límite de una cuenta, y es fácil pasarlos por alto. Al diseñar su modelo de autorización, asegúrese de tener en cuenta estas suposiciones implícitas para que puedan documentarse y representarse formalmente en el modelo de autorización.

Separar las entidades principales de los contenedores de recursos

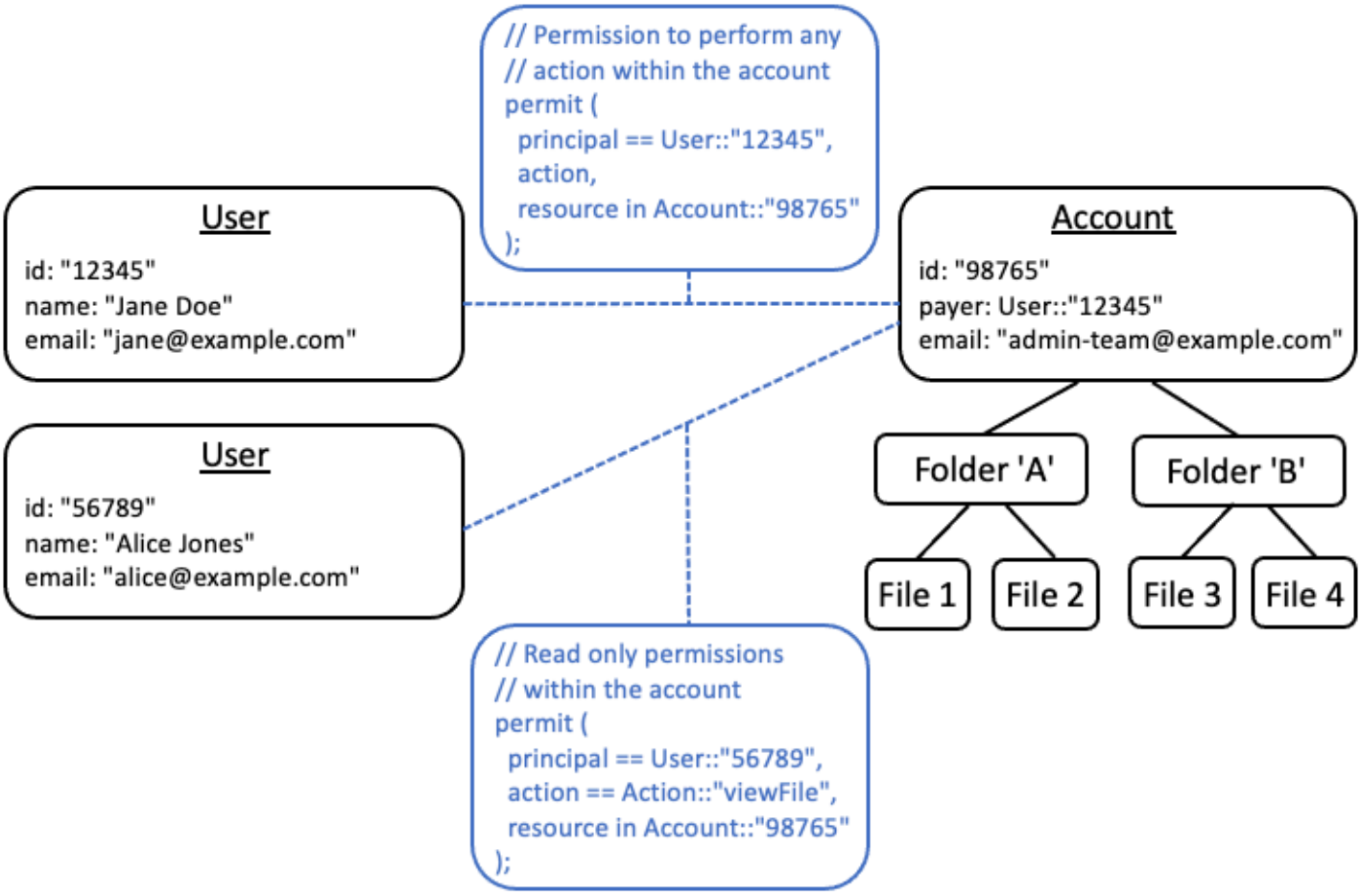
Al diseñar una jerarquía de recursos, una de las tendencias más comunes, especialmente en el caso de las aplicaciones orientadas al consumidor, es utilizar la identidad de usuario del cliente como contenedor de recursos en una cuenta de cliente.



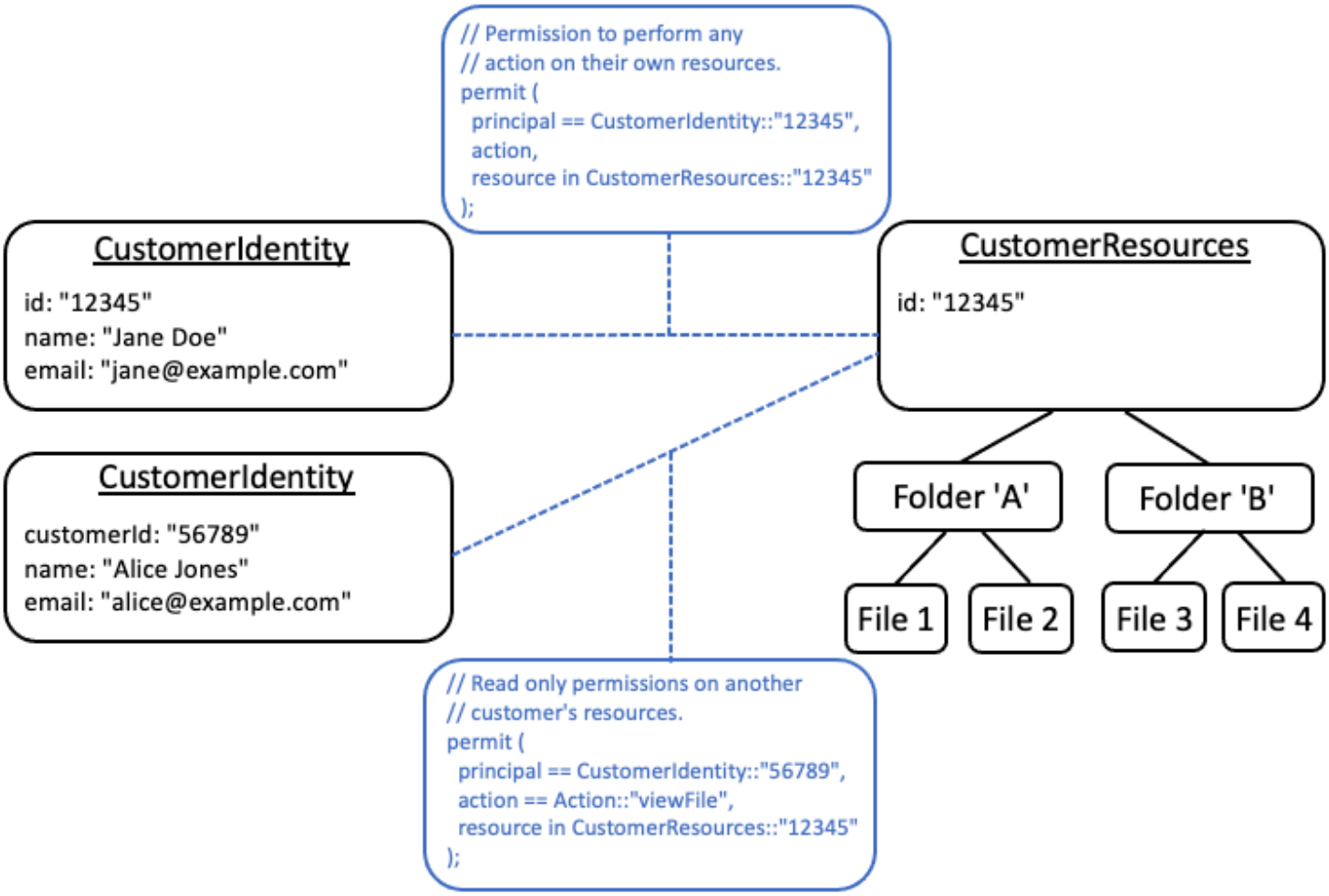
Le recomendamos que trate esta estrategia como un antipatrón. Esto se debe a que existe una tendencia natural en las aplicaciones con más funciones a delegar el acceso a usuarios adicionales. Por ejemplo, puede optar por introducir cuentas “familiares”, en las que otros usuarios puedan compartir los recursos de la cuenta. Del mismo modo, los clientes empresariales a veces desean designar a varios miembros de la plantilla como operadores de algunas partes de la cuenta. Es posible que también necesite transferir la propiedad de una cuenta a otro usuario o combinar los recursos de varias cuentas.

Cuando se utiliza la identidad de un usuario como contenedor de recursos para una cuenta, resulta más difícil tener éxito en los escenarios anteriores. Lo que es más preocupante es que, si se concede a otras personas acceso al contenedor de la cuenta con este enfoque, es posible que se les conceda acceso de manera inadvertida para modificar la propia identidad del usuario; por ejemplo, cambiando el correo electrónico o las credenciales de inicio de sesión de Jane.

Por lo tanto, siempre y cuando sea posible, un enfoque más flexible consiste en separar las entidades principales de los contenedores de recursos y modelar la conexión entre ellos utilizando conceptos como “permisos de administrador” o “propiedad”.



Si tiene una aplicación que no puede seguir este modelo disociado, le recomendamos que considere la posibilidad de imitarlo en la medida de lo posible cuando diseñe un modelo de autorización. Por ejemplo, una aplicación que posea un solo concepto denominado `Customer` que encapsule la identidad del usuario, las credenciales de inicio de sesión y los recursos que posee podría asignarlo a un modelo de autorización que contenga una entidad lógica para `Customer Identity` (que contenga el nombre, el correo electrónico, etc.) y una entidad lógica independiente para `Customer Resources` o `Customer Account`, que actúe como nodo principal para todos los recursos que posee. Ambas entidades pueden compartir el mismo `Id`, pero con un`Type` diferente.



Uso de atributos o plantillas para representar relaciones

Hay dos formas principales de expresar las relaciones entre los recursos. El momento de utilizar una u otra depende de si la relación ya está almacenada en la base de datos de la aplicación o no y se utiliza por otros motivos, como el cumplimiento. Si es así, adopte el enfoque basado en atributos. Si no es así, entonces adopte el enfoque basado en plantillas.

Relaciones basadas en atributos

Los atributos se pueden utilizar como entrada en la decisión de autorización para representar una relación entre un principal y uno o más recursos.

Este patrón es adecuado cuando se hace un seguimiento de la relación y se administra con fines que van más allá de la mera administración de permisos. Por ejemplo, es necesario registrar al titular principal de la cuenta para cumplir desde el punto de vista financiero con las normas de Conozca a

su cliente. Los permisos se derivan de estas relaciones. Los datos de la relación se administran fuera del sistema de autorización y se obtienen como entrada al tomar una decisión de autorización.

El siguiente ejemplo muestra cómo se puede representar la relación entre un usuario `Alice` y varias cuentas de las que es el titular principal de la cuenta:

```
// Using a user attribute to represent the primary account holder relationship
{
  "id": "df82e4ad-949e-44cb-8acf-2d1acda71798",
  "name": "alice",
  "email": "alice@example.com",
  "primaryOnAccounts": [
    "Account::\"c943927f-d803-4f40-9a53-7740272cb969\"",
    "Account::\"b8ee140c-fa09-46c3-992e-099438930894\""
  ]
}
```

Y, posteriormente, utilizar el atributo dentro de una política:

```
// Derived relationship permissions
  permit (
    principal,
    action in Action::"primaryAccountHolderActions",
    resource
  )when {
    resource in principal.primaryOnAccounts
  };
```

Por el contrario, la misma relación podría representarse como un atributo del recurso denominado `primaryAccountHolders` que contiene un conjunto de usuarios.

Si hay varios tipos de relaciones entre los principales y los recursos, estos se deben modelar como atributos diferentes. Por ejemplo, si las cuentas también pueden tener firmantes autorizados y estas personas tienen permisos diferentes en la cuenta, esto se representaría como un atributo diferente.

En el caso anterior, también `Alice` podría ser un firmante autorizado en una tercera cuenta. El siguiente ejemplo muestra cómo se podría representar esto:

```
// Using user attributes to represent the primary account holder and authorized
 signatory relationships
{
  "id": "df82e4ad-949e-44cb-8acf-2d1acda71798",
```

```
  "name": "alice",
  "email": "alice@example.com",
  "primaryOnAccounts": [
    "Account::\"c943927f-d803-4f40-9a53-7740272cb969\"",
    "Account::\"b8ee140c-fa09-46c3-992e-099438930894\""
  ],
  "authorizedSignatoryOnAccounts": [
    "Account::\"661817a9-d478-4096-943d-4ef1e082d19a\""
  ]
}
```

Las políticas correspondientes son las siguientes:

```
// Derived relationship permissions

  permit (
    principal,
    action in Action::"primaryAccountHolderActions",
    resource
  )when {
    resource in principal.primaryOnAccounts
  };
  permit (
    principal,
    action in Action::"authorizedSignatoryActions",
    resource
  )when {
    resource in principal.authorizedSignatoryOnAccounts
  };
```

Relaciones basadas en plantillas

Si la relación entre los recursos existe únicamente con el propósito de administrar los permisos, es conveniente almacenar esta relación como una política o plantilla vinculada a una plantilla. También puedes pensar en estas plantillas como funciones que se asignan a un recurso específico.

Por ejemplo, en un sistema de administración de documentos`Alice`, el propietario del documento puede optar por conceder permiso a otro usuario para contribuir al documento. `Bob` Esto establece una relación de colaboración entre Bob y el documento de Alice. El único propósito de esta relación es conceder permiso para editar y comentar el documento y, por lo tanto, esta relación se puede representar como una plantilla. En estos casos, el enfoque recomendado es crear una plantilla

para cada tipo de relación. En los ejemplos siguientes hay dos tipos de relaciones `Contributor` y`Reviewer`, por lo tanto, dos plantillas.

Las siguientes plantillas se pueden utilizar para crear políticas vinculadas a plantillas para usuarios individuales.

```
// Managed relationship permissions - Contributor template
permit (
  principal == ?principal,
  action in Action::"DocumentContributorActions",
  resource in ?resource
);

// Managed relationship permissions - Reviewer template
permit (
  principal == ?principal,
  action in Action::"DocumentReviewerActions",
  resource in ?resource
);
```

Las siguientes plantillas se pueden utilizar para crear políticas vinculadas a plantillas para grupos de usuarios. La única diferencia con las plantillas para usuarios individuales es que utilizan el `in` operador en lugar del. ==

```
// Managed relationship permissions - Contributor template
permit (
  principal in ?principal,
  action in Action::"DocumentContributorActions",
  resource in ?resource
);

// Managed relationship permissions - Reviewer template
permit (
  principal in ?principal,
  action in Action::"DocumentReviewerActions",
  resource in ?resource
);
```

A continuación, puede utilizar estas plantillas para crear políticas, como las siguientes, que representen los permisos de relaciones gestionadas cada vez que se concede acceso a un documento.

```
//Managed relationship permissions
permit (
  principal in User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
  action in Action::"DocumentContributorActions",
  resource in Document::"c943927f-d803-4f40-9a53-7740272cb969"
);

permit (
  principal in UserGroup::"df82e4ad-949e-44cb-8acf-2d1acda71798",
  action in Action::"DocumentReviewerActions",
  resource == Document::"661817a9-d478-4096-943d-4ef1e082d19a"
);

permit (
  principal in User::"df82e4ad-949e-44cb-8acf-2d1acda71798",
  action in Action::"DocumentContributorActions",
  resource in Folder::"b8ee140c-fa09-46c3-992e-099438930894"
);
```

Amazon Verified Permissions puede gestionar de manera eficiente muchas políticas individuales y detalladas durante la evaluación de la autorización y modelar las cosas de esta manera significa que Verified Permissions mantiene un registro de auditoría completo de todas las AWS CloudTrail decisiones de autorización.

Preferencia por los permisos detallados en el modelo y los permisos agregados en la interfaz de usuario

Una estrategia de la que los diseñadores suelen arrepentirse más tarde es diseñar un modelo de autorización con acciones muy amplias, como `Read` y `Write`, y darse cuenta más tarde de que es necesario adoptar medidas más específicas. La necesidad de un nivel más detallado puede estar motivada por los comentarios de los clientes a favor de controles de acceso más precisos o por los auditores de cumplimiento y seguridad, que recomiendan el uso de permisos con privilegios mínimos.

Si los permisos específicos no se definen por adelantado, tal vez se necesite una compleja conversión para cambiar el código de la aplicación y las instrucciones de política por permisos de usuario más específicos. Por ejemplo, será necesario modificar un código de la aplicación que anteriormente autorizaba una acción amplia para que utilice las acciones detalladas. Además, las políticas deberán actualizarse para reflejar la migración:

```
permit (
    principal == User::"6688f676-1aa9-456a-acf4-228340b54e9d",
    // action  ==  Action::"read",              -- coarse-grained permission --
 commented out
    action in [                        //        -- finer grained permissions
        Action::"listFolderContents",
        Action::"viewFile"
    ],
    resource in Account::"c863f89b-461f-4fc2-b638-e5fa5f79a48b"
);
```

Para evitar esta costosa migración, es mejor definir los permisos detallados por adelantado. Sin embargo, esto puede resultar en una desventaja si, posteriormente, los usuarios finales se ven obligados a comprender un mayor número de permisos detallados, especialmente si la mayoría de los clientes estarían satisfechos con controles generales, como `Read` y `Write`. Para quedarse con lo mejor de ambos mundos, puede agrupar los permisos detallados en colecciones predefinidas, como `Read` y `Write`, y utilizar mecanismos como plantillas de políticas o grupos de acciones. Al usar este enfoque, los clientes solo ven los permisos generales. Sin embargo, entre bastidores, ha preparado su aplicación para el futuro al modelar los permisos generales como un conjunto de acciones detalladas. Cuando los clientes o los auditores lo soliciten, se pueden exponer los permisos detallados.

Consideración de otros motivos para solicitar la autorización

Por lo general, asociamos las comprobaciones de autorización con las solicitudes de los usuarios. La verificación es una forma de determinar si el usuario tiene permiso para realizar esa solicitud. Sin embargo, también puede utilizar los datos de autorización como un factor que influye en el diseño de la interfaz de la aplicación. Por ejemplo, es posible que desee mostrar una pantalla de inicio con una lista únicamente de los recursos a los que puede acceder el usuario final. Cuando consulte los detalles de un recurso, es posible que desee que la interfaz muestre solo las operaciones que el usuario puede realizar en ese recurso.

Estas situaciones pueden introducir desventajas en el modelo de autorización. Por ejemplo, depender en gran medida de las políticas de control de acceso basadas en atributos (ABAC) puede dificultar la respuesta rápida a la pregunta «¿quién tiene acceso a qué?» Esto se debe a que responder a esa pregunta requiere examinar cada regla comparándola con todas las entidades principales y recursos para determinar si coinciden. Como resultado, un producto que necesite optimizarse para incluir solo los recursos a los que puede acceder el usuario podría optar por utilizar

un modelo de control de acceso basado en roles (). RBAC Al usarloRBAC, puede resultar más fácil iterar todas las políticas asociadas a un usuario para determinar el acceso a los recursos.

Almacenes de políticas de Amazon Verified Permissions

Un almacén de políticas es un contenedor de políticas y plantillas de políticas. En cada almacén de políticas, puede crear un esquema que se utilice para validar las políticas añadidas al almacén de políticas. Además, puede activar la validación de políticas. Si agrega una política a un almacén de políticas con la validación de políticas habilitada, los tipos de entidades, los tipos comunes y las acciones definidas en la política se validan con el esquema y las políticas no válidas se rechazan.

Se recomienda crear un almacén de políticas por aplicación o un almacén de políticas por inquilino para las aplicaciones de varios inquilinos. Debe especificar un almacén de políticas al realizar una solicitud de autorización.

Le recomendamos que utilice espacios de nombres para las entidades de Cedar en sus almacenes de políticas para evitar la ambigüedad. Un espacio de nombres es un prefijo de cadena para un tipo, separado por un par de signos de dos puntos (`::`) como delimitador. Verified Permissions admite solo un espacio de nombres por almacén de políticas. Estos espacios de nombres ayudan a mantener las cosas claras cuando trabajas con varias aplicaciones similares. Por ejemplo, en las aplicaciones con varios inquilinos, si se utiliza un espacio de nombres para añadir el nombre del inquilino a los tipos definidos en el esquema, se diferenciarán de sus homólogos similares utilizados por los demás inquilinos. Al consultar los registros de las solicitudes de autorización, podrá identificar fácilmente al inquilino que procesó la solicitud de autorización. Para obtener información más detallada consulte Espacios de nombres en la Guía de referencia sobre el lenguaje de las políticas de Cedar.

Temas

- Crear almacenes de políticas de Verified Permissions
- API-tiendas de pólizas vinculadas
- Eliminar almacenes de políticas

Crear almacenes de políticas de Verified Permissions

Puede crear un almacén de políticas mediante uno de los siguientes métodos:

- Siga una configuración guiada: definirá un tipo de recurso con acciones válidas y un tipo principal antes de crear su primera política.

- Configure con API Gateway y una fuente de identidad: defina sus entidades principales con los usuarios que inician sesión con un proveedor de identidad (IdP) y sus acciones y entidades de recursos desde un Amazon API Gateway. API Le recomendamos esta opción si desea que su aplicación autorice API las solicitudes relacionadas con la pertenencia a un grupo de usuarios.
- Comience con un ejemplo de almacén de políticas: elija un ejemplo de almacén de políticas de proyecto predefinido. Le recomendamos esta opción si está aprendiendo sobre Verified Permissions y quiere ver y probar ejemplos de políticas.
- Cree un almacén de políticas vacío: definirá usted mismo el esquema y todas las políticas de acceso. Recomendamos esta opción si ya está familiarizado con la configuración de un almacén de políticas.

Guided setup

Para crear un almacén de políticas con el método Configuración guiada

El asistente de configuración guiada le guiará por el proceso de creación de la primera iteración de su almacén de políticas. Creará un esquema para el primer tipo de recurso, describirá las acciones que se aplican a ese tipo de recurso y el tipo de entidad principal para el que va a conceder permisos. A continuación, creará su primera política. Una vez que haya completado este asistente, podrá agregarla a su almacén de políticas, ampliar el esquema para describir otros tipos de recursos y entidades principales y crear políticas y plantillas adicionales.

1. En la consola de permisos verificados, seleccione Crear un nuevo almacén de políticas.
2. En la sección Opciones de inicio, selecciona Configuración guiada.
3. Introduzca una descripción del almacén de políticas. Este texto puede ser el que mejor se adapte a su organización como referencia sencilla a la función del almacén de políticas actual, por ejemplo, las actualizaciones meteorológicas.
4. En la sección Detalles, escriba un espacio de nombres para su esquema.
5. Elija Next (Siguiente).
6. En la ventana Tipo de recurso, escriba un nombre para el tipo de recurso.
7. (Opcional) Seleccione Agregar un atributo para añadir los atributos del recurso. Escriba el nombre del atributo y seleccione un tipo de atributo para cada atributo del recurso. Elija si cada atributo es obligatorio. Verified Permissions utiliza los valores de atributo especificados al verificar las políticas con el esquema. Para eliminar un atributo que se ha añadido al tipo de recurso, seleccione Eliminar junto al atributo.

8. En el campo Acciones, escriba las acciones que se van a autorizar para el tipo de recurso especificado. Para agregar acciones adicionales para el tipo de recurso, elija Agregar una acción. Para eliminar una acción que se ha añadido al tipo de recurso, seleccione Eliminar junto a la acción.
9. En el campo Nombre del tipo de entidad principal, escriba el nombre del tipo de entidad principal que utilizará las acciones especificadas para el tipo de recurso.
10. Elija Next (Siguiente).
11. En la ventana Tipo de entidad principal, elija la fuente de identidad para su tipo de entidad principal.

    - Elija Personalizado si la aplicación de Verified Permissions proporcionará directamente el ID y los atributos de la entidad principal. Para añadir atributos a la entidad principal, elija Agregar un atributo. Escriba el nombre del atributo y seleccione un tipo de atributo para cada atributo de la entidad principal. Verified Permissions utiliza los valores de atributo especificados al verificar las políticas con el esquema. Para eliminar un atributo que se ha añadido al tipo de entidad principal, seleccione Eliminar junto al atributo.
    - Elija Grupo de usuarios de Cognito si el ID y los atributos de la entidad principal se proporcionarán a partir de un identificador o un token de acceso generado por Amazon Cognito. Seleccione Conectar grupo de usuarios. Seleccione la Región de AWSy escriba el ID del grupo de usuarios de Amazon Cognito al que desea conectarse. Elija Conectar. Para obtener más información, consulte Autorización con Amazon Verified Permissions en la Guía para desarrolladores de Amazon Cognito.
12. Elija Next (Siguiente).
13. En la sección Detalles de la política, escriba una descripción de la política opcional para su primera política de Cedar.
14. En el campo Ámbito de las entidades principales, elija las entidades principales a las que se les concederán los permisos de la política.

    - Elija Entidad principal específica para aplicar la política a una entidad principal concreta. Elija la entidad principal en el campo Entidad principal a la que se permitirá realizar acciones y escriba un identificador de entidad para la entidad principal.
    - Seleccione Todas las entidades principales para aplicar la política a todas las entidades principales de su almacén de políticas.
15. En el campo Ámbito de los recursos, elija los recursos sobre los que las entidades principales especificadas tendrán autorización para actuar.

- Seleccione Recurso específico para aplicar la política a un recurso específico. Elija el recurso en el campo Recurso al que se debe aplicar esta política y escriba un identificador de entidad para el recurso.
- Seleccione Todos los recursos para aplicar la política a todos los recursos de su almacén de políticas.

16. En el campo Ámbito de las acciones, elija las acciones para las que las entidades principales especificadas tendrán autorización para llevar a cabo.

    - Seleccione Conjunto específico de acciones para aplicar la política a acciones concretas. Seleccione las casillas de verificación situadas junto al campo Acciones a las que se debe aplicar esta política.
    - Seleccione Todas las acciones para aplicar la política a todas las acciones de su almacén de políticas.

17. Revise la política en la sección Vista previa de la política. Seleccione Crear almacén de políticas.

Set up with API Gateway and an identity source

Para crear un almacén de políticas mediante el método de configuración Configurar con API Gateway y una fuente de identidad

La opción API Gateway se protege APIs con políticas de permisos verificados que están diseñadas para tomar decisiones de autorización a partir de los grupos o roles de los usuarios. Esta opción crea un almacén de políticas para probar la autorización con grupos de fuentes de identidad y uno API con un autorizador Lambda.

Los usuarios y sus grupos de un IdP se convierten en sus directores (identificadores) o en su contexto (identificadores de acceso). Los métodos y las rutas de un API Gateway API se convierten en las acciones que autorizan sus políticas. Su aplicación se convierte en el recurso. Como resultado de este flujo de trabajo, Verified Permissions crea un almacén de políticas, una función Lambda y un autorizador LambdaAPI. Debe asignarle el autorizador Lambda cuando API termine este flujo de trabajo.

1. En la consola de permisos verificados, seleccione Crear un nuevo almacén de políticas.
2. En la sección Opciones de inicio, elija Configurar con API Gateway y una fuente de identidad y, a continuación, seleccione Siguiente.

3. En el paso Importar recursos y acciones API, elija uno API que sirva de modelo para los recursos y acciones del almacén de políticas.

   a. Elija una etapa de despliegue de las etapas configuradas en la suya API y seleccione Importar API. Para obtener más información sobre API las etapas, consulte Configuración de una etapa para una REST API en la Guía para desarrolladores de Amazon API Gateway.

   b. Obtenga una vista previa del mapa con los recursos y acciones importados.

   c. Para actualizar los recursos o las acciones, modifique sus API rutas o métodos y seleccione Importar API.

   d. Cuando esté satisfecho con sus opciones, elija Siguiente.

4. En Origen de identidad, elija un tipo de proveedor de identidad. Puede elegir un grupo de usuarios de Amazon Cognito o un tipo de IdP de OpenID Connect (). OIDC

5. Si eligió Amazon Cognito:

   a. Elija un grupo de usuarios en el mismo almacén de políticas Región de AWS y en el Cuenta de AWS que se encuentre.

   b. Elija el tipo de token al API que desea transferir y que desea enviar para su autorización. Ambos tipos de token contienen grupos de usuarios, que son la base de este modelo API de autorización vinculada.

   c. En la sección Validación de clientes de aplicaciones, puede limitar el alcance de un almacén de políticas a un subconjunto de los clientes de la aplicación Amazon Cognito en un grupo de usuarios de varios inquilinos. Para solicitar que el usuario se autentique con uno o más clientes de aplicaciones específicos de su grupo de usuarios, seleccione Aceptar solo los tokens con el cliente de aplicación esperado. IDs Para aceptar a cualquier usuario que se autentique en el grupo de usuarios, selecciona No validar el cliente de la aplicación. IDs

   d. Elija Next (Siguiente).

6. Si has elegido un OIDCproveedor:

   a. En Emisor URL, introduzca el URL de su OIDC emisor. Este es el punto final del servicio que proporciona el servidor de autorización, las claves de firma y otra información sobre su proveedor, por ejemplo. `https://auth.example.com` El emisor URL debe alojar un documento de OIDC descubrimiento en`/.well-known/openid-configuration`.

b. En Tipo de token, elija el tipo OIDC JWT que desea que envíe su solicitud de autorización. Para obtener más información, consulte Asignación de tokens de proveedores de identidad al esquema.

c. En Notificaciones de token, elige cómo quieres configurar los atributos de usuario en tu almacén de políticas. Estos atributos definen las afirmaciones a las que pueden hacer referencia tus políticas.

   i. Elige una fuente de reclamación.

      A. Para proporcionar un token de muestra, selecciona Extraer de la JWT carga útil y pega la carga útil de un token del tipo JWT de token que hayas elegido. JWTscontienen un encabezado, una carga útil y una firma. La muestra JWT debe estar decodificada y solo debe cargarse. Para analizar la carga útil, selecciona Extraer.

      B. Para introducir tu propio conjunto de atributos, selecciona Introducir las reclamaciones manualmente.

   ii. Introduzca o confirme el nombre y el tipo de valor de cada reclamación de token que desee añadir a los atributos del contexto principal o de acción del usuario en su esquema.

d. En las notificaciones de usuario y grupo, elija una afirmación de usuario para la fuente de identidad. Por lo general`sub`, se trata de una afirmación que proviene de tu ID o token de acceso que contiene el identificador único de la entidad que se va a evaluar. Las identidades del OIDC IdP conectado se asignarán al tipo de usuario del almacén de políticas.

e. En Notificaciones de usuarios y grupos, elija una notificación de grupo para la fuente de identidad. Por lo general`groups`, se trata de una afirmación de tu ID o token de acceso que contiene una lista de los grupos del usuario. El almacén de políticas autorizará las solicitudes en función de la pertenencia al grupo.

f. En Validación de audiencia o Cliente IDs, introduzca el cliente IDs o público URLs que desea que su almacén de políticas acepte en las solicitudes de autorización, si las hubiera. En el caso de los tokens de acceso, introduce un valor de reclamación de audiencia, como`https://myapp.example.com`. En el caso de los identificadores, introduce un identificador de cliente similar`1example23456789`.

g. Elija Next (Siguiente).

7. Si ha elegido Amazon Cognito, Verified Permissions consulta los grupos de usuarios. En el caso OIDC de los proveedores, introduzca los nombres de los grupos manualmente. El paso Asignar acciones a los grupos crea políticas para el almacén de políticas que permiten a los miembros del grupo realizar acciones.

    a. Elija o añada los grupos que desee incluir en sus políticas.

    b. Asigna acciones a cada uno de los grupos que has seleccionado.

    c. Elija Next (Siguiente).

8. En Implementar la integración de aplicaciones, revise los pasos que Verified Permissions realizará para crear el almacén de políticas y el autorizador de Lambda.

9. Cuando esté listo para crear los nuevos recursos, elija Crear e implementar.

10. Mantén abierto el paso de estado del almacén de políticas en tu navegador para supervisar el progreso de la creación de recursos mediante permisos verificados.

11. Después de algún tiempo, normalmente alrededor de una hora, o cuando el paso Implementar el autorizador Lambda muestre éxito, configure el autorizador.

    Los permisos verificados habrán creado una función Lambda y un autorizador Lambda en su. API Seleccione Abrir API para ir a su. API

    Para obtener información sobre cómo asignar un autorizador Lambda, consulte Uso de autorizadores API Lambda de Gateway en la Guía para desarrolladores de Amazon Gateway. API

    a. Diríjase a Autorizadores API y anote el nombre del autorizador que creó Verified Permissions.

    b. Ve a Recursos y selecciona un método de nivel superior en tu. API

    c. Selecciona Editar en la configuración de solicitud de métodos.

    d. Configure el autorizador para que sea el nombre del autorizador que anotó anteriormente.

    e. Amplíe los encabezados de las HTTP solicitudes, introduzca un nombre o y seleccione **`AUTHORIZATION`** Obligatorio.

    f. Despliegue el API escenario.

    g. Guarde los cambios.

12. Pruebe su autorizador con un token de grupo de usuarios del tipo de token que seleccionó en el paso Elegir la fuente de identidad. Para obtener más información sobre el inicio de sesión

del grupo de usuarios y la recuperación de los tokens, consulte el flujo de autenticación del grupo de usuarios en la Guía para desarrolladores de Amazon Cognito.

13. Vuelva a probar la autenticación con un token de grupo de usuarios en el `AUTHORIZATION` encabezado de la solicitud que se le envíe. API
14. Examine su nuevo almacén de políticas. Añada y perfeccione las políticas.

Sample policy store

Para crear un almacén de políticas con el método de configuración Almacén de políticas de muestra

1. En la sección Opciones de inicio, selecciona un almacén de políticas de muestra.
2. En la sección Ejemplo de proyecto, elija el tipo de aplicación de Verified Permissions de muestra que va a utilizar.

   - PhotoFlashes un ejemplo de aplicación web orientada al cliente que permite a los usuarios compartir fotos y álbumes individuales con amigos. Los usuarios pueden establecer permisos detallados sobre quién puede ver, comentar y volver a compartir sus fotos. Los propietarios de las cuentas también pueden crear grupos de amigos y organizar las fotos en álbumes.
   - DigitalPetStorees un ejemplo de aplicación en el que cualquiera puede registrarse y convertirse en cliente. Los clientes pueden añadir mascotas para vender, buscar mascotas y realizar pedidos. Los clientes que han añadido una mascota se registran como propietarios de la mascota. Los dueños de mascotas pueden actualizar los detalles de la mascota, subir una imagen de ella o eliminar la lista de mascotas. Los clientes que han realizado un pedido quedan registrados como propietarios del pedido. Los propietarios de los pedidos pueden obtener los detalles del pedido o cancelarlo. Los gerentes de las tiendas de mascotas tienen acceso de administrador.

     > **Note**
     >
     > El almacén de políticas de DigitalPetStoremuestra no incluye plantillas de políticas. Los almacenes TinyTododе políticas PhotoFlashy los de muestra incluyen plantillas de políticas.

- TinyTodoes una aplicación de ejemplo que permite a los usuarios crear tareas y listas de tareas. Los propietarios de las listas pueden administrar y compartir sus listas y especificar quién puede verlas o editarlas.

3. Se generará automáticamente un espacio de nombres para el esquema del almacén de políticas de muestra en función del proyecto de ejemplo que haya elegido.
4. Seleccione Crear almacén de políticas.

   El almacén de políticas se crea con políticas y un esquema para el almacén de políticas de muestra que elija. Para obtener más información sobre las políticas vinculadas a plantillas que puede crear para los almacenes de políticas de muestra, consulte Ejemplos de políticas vinculadas a plantillas de permisos verificados de Amazon.

Empty policy store

Para crear un almacén de políticas con el método de configuración Almacén de políticas vacío

1. En la sección Opciones de inicio, elija Vacía el almacén de políticas.
2. Seleccione Crear almacén de políticas.

Se crea un almacén de políticas vacío sin un esquema, lo que significa que las políticas no se validan. Para obtener más información acerca de cómo actualizar el esquema del almacén de políticas, consulte Esquema del almacén de políticas de Amazon Verified Permissions..

Para obtener más información sobre cómo crear políticas para su almacén de políticas, consulte Creación de políticas estáticas de Amazon Verified Permissions y Creación de políticas vinculadas a plantillas de permisos verificados de Amazon.

AWS CLI

Para crear un almacén de políticas vacío mediante la AWS CLI.

Puede crear un almacén de políticas mediante la operación `create-policy-store`.

> **Note**
>
> Un almacén de políticas que se crea mediante el AWS CLI está vacío.
>
> - Para añadir un esquema, consulte Esquema del almacén de políticas de Amazon Verified Permissions..

- Para añadir políticas, consulte Creación de políticas estáticas de Amazon Verified Permissions.
- Para añadir plantillas de políticas, consulte Creación de plantillas de políticas de permisos verificados de Amazon.

```
$ aws verifiedpermissions create-policy-store \
    --validation-settings "mode=STRICT"
{
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T17:41:29.103459+00:00",
    "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

AWS SDKs

Puede crear un almacén de políticas mediante `CreatePolicyStore`API. Para obtener más información, consulta CreatePolicyStorela Guía de API referencia de permisos verificados de Amazon.

API-tiendas de pólizas vinculadas

Al crear un nuevo almacén de políticas en la consola de permisos verificados de Amazon, puede elegir la opción Configurar con API Gateway y una fuente de identidad. Con esta opción, puede crear un almacén API de políticas vinculado, un modelo de autorización para aplicaciones que se autentican con grupos de usuarios de Amazon Cognito o con OIDC un proveedor de identidad (IdP) y obtiene datos de Amazon Gateway. API APIs Para empezar, consulte Crear un almacén de políticas para usar API Gateway con un proveedor de identidades.

Temas

- Cómo autoriza Verified Permissions las solicitudes API
- Consideraciones sobre los API almacenes de pólizas vinculados
- Añadir un control de acceso basado en atributos () ABAC
- Pasar a la producción con AWS CloudFormation
- Solución de problemas de almacenes de políticas vinculados API

> **Important**
>
> Los almacenes de políticas que cree con la opción Configurar con API Gateway y una fuente de identidad en la consola de permisos verificados no están pensados para su implementación inmediata en producción. Con el almacén de políticas inicial, finalice el modelo de autorización y exporte los recursos del almacén de políticas al CloudFormation. Implemente permisos verificados en producción mediante programación con el AWSCloud Development Kit () CDK. Para obtener más información, consulte Pasar a la producción con AWS CloudFormation.

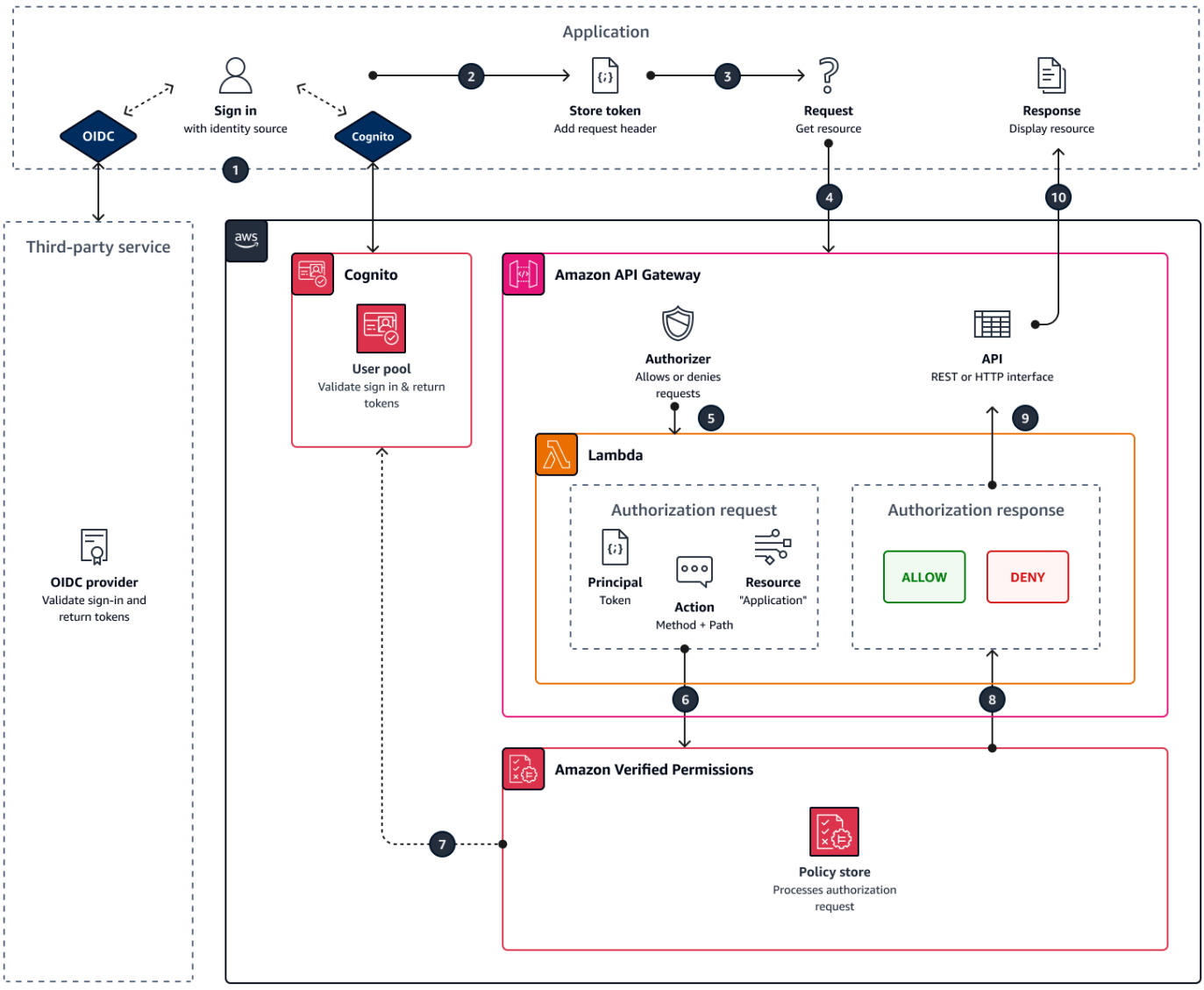
En un almacén de políticas que está vinculado a una fuente de identidad API y a una fuente de identidad, tu aplicación presenta un token de grupo de usuarios en un encabezado de autorización cuando realiza una solicitud a. API La fuente de identidad de tu almacén de políticas proporciona una validación de token para los permisos verificados. El token forma la entrada `principal` de las solicitudes de autorización con IsAuthorizedWithTokenAPI. Los permisos verificados crean políticas en torno a la pertenencia a grupos de sus usuarios, tal como se presenta en la declaración de un grupo en términos de identidad (ID) y de acceso, por ejemplo, `cognito:groups` para los grupos de usuarios. APIProcesa el token de la aplicación en un autorizador Lambda y lo envía a Verified Permissions para que tome una decisión de autorización. Cuando API reciba la decisión de autorización del autorizador Lambda, este transferirá la solicitud a su fuente de datos o la rechazará.

Componentes de la fuente de identidad y la autorización de API Gateway con permisos verificados

- Un grupo de usuarios o OIDC IdP de Amazon Cognito que autentica y agrupa a los usuarios. Los tokens de los usuarios completan la membresía del grupo y el principal o el contexto que Verified Permissions evalúa en su almacén de políticas.
- Una API puerta de enlace. REST API Los permisos verificados definen las acciones a partir de API rutas y API métodos, por ejemplo`MyAPI::Action::get /photo`.
- Una función Lambda y un autorizador Lambda para su. API La función Lambda toma los tokens portadores de su grupo de usuarios, solicita la autorización de Verified Permissions y devuelve una decisión a Gateway. API El flujo de trabajo Configurar con Cognito y API Gateway crea automáticamente este autorizador Lambda por usted.
- Un almacén de políticas de permisos verificados. La fuente de identidad del almacén de políticas es su grupo de usuarios. El esquema del almacén de políticas refleja su API configuración y las políticas vinculan a los grupos de usuarios con API las acciones permitidas.
- Una aplicación que autentica a los usuarios con tu IDP y añade tokens a las solicitudes. API

Cómo autoriza Verified Permissions las solicitudes API

Al crear un nuevo almacén de políticas y seleccionar la opción Configurar con Cognito y API Gateway, Verified Permissions crea el esquema y las políticas del almacén de políticas. El esquema y las políticas reflejan API las acciones y los grupos de grupos de usuarios que desea autorizar para que las realicen. Verified Permissions también crea la función Lambda y el autorizador. Debe configurar el nuevo autorizador en un método de su. API



1. El usuario inicia sesión con la aplicación a través de Amazon Cognito u otro OIDC IdP. El IdP emite identificadores de acceso y de identificación con la información del usuario.

2. Su aplicación almacena elJWTs. Para obtener más información, consulte Uso de tokens con grupos de usuarios en la Guía para desarrolladores de Amazon Cognito.
3. El usuario solicita datos que la aplicación debe recuperar de un dispositivo externoAPI.
4. Su aplicación solicita datos de una API puerta REST API de enlace interna. Añade un ID o un token de acceso como encabezado de solicitud.
5. Si API tiene una memoria caché para la decisión de autorización, devolverá la respuesta anterior. Si el almacenamiento en caché está deshabilitado o no API tiene caché actual, API Gateway pasa los parámetros de la solicitud a un autorizador Lambda basado en tokens.
6. La función Lambda envía una solicitud de autorización a un almacén de políticas de permisos verificados con. IsAuthorizedWithTokenAPI La función Lambda transmite los elementos de una decisión de autorización:
   a. El token del usuario como principal.
   b. El API método combinado con la API ruta, por ejemplo`GetPhoto`, como acción.
   c. El término `Application` como recurso.
7. Los permisos verificados validan el token. Para obtener más información sobre cómo se validan los tokens de Amazon Cognito, consulte Autorización con permisos verificados de Amazon en la Guía para desarrolladores de Amazon Cognito.
8. Verified Permissions evalúa la solicitud de autorización comparándola con las políticas de su almacén de políticas y devuelve una decisión de autorización.
9. El autorizador Lambda devuelve una `Deny` respuesta `Allow` or a Gateway. API
10. APIDevuelve datos o una `ACCESS_DENIED` respuesta a su solicitud. Su solicitud procesa y muestra los resultados de la API solicitud.

Consideraciones sobre los API almacenes de pólizas vinculados

Al crear un almacén API de políticas vinculado a puntos en la consola de permisos verificados, se crea una prueba para una posible implementación en producción. Antes de pasar a la fase de producción, establezca una configuración fija para su grupo de usuarios API y su grupo de usuarios. Tenga en cuenta los siguientes factores:

APIGateway almacena en caché las respuestas

En los almacenes API de políticas enlazados, Verified Permissions crea un autorizador Lambda con un almacenamiento en TTL caché de autorización de 120 segundos. Puede ajustar este valor o desactivar el almacenamiento en caché en su autorizador. En un autorizador con el

almacenamiento en caché activado, el autorizador devuelve la misma respuesta cada vez hasta que caduca. TTL Esto puede prolongar la vida útil efectiva de los tokens del grupo de usuarios hasta un tiempo equivalente al almacenamiento en caché TTL de la fase solicitada.

Los grupos de Amazon Cognito se pueden reutilizar

Amazon Verified Permissions determina la pertenencia a un grupo para los usuarios del grupo de usuarios a partir de la `cognito:groups` declaración que figura en el identificador de usuario o en el token de acceso. El valor de esta afirmación es un conjunto de nombres descriptivos de los grupos de usuarios a los que pertenece el usuario. No puede asociar grupos de grupos de usuarios con un identificador único.

Los grupos de usuarios que se eliminan y se vuelven a crear con el mismo nombre se presentan en el almacén de políticas como el mismo grupo. Al eliminar un grupo de un grupo de usuarios, elimine todas las referencias al grupo del almacén de políticas.

API-el espacio de nombres y el esquema derivados son point-in-time

Verified Permissions lo captura en un API momento dado: solo lo consulta API cuando crea su almacén de políticas. Cuando el esquema o el nombre del usuario API cambien, debe actualizar el almacén de políticas y el autorizador de Lambda, o bien crear un nuevo almacén API de políticas vinculado. Verified Permissions deriva el espacio de nombres del almacén de políticas a partir del nombre suyo. API

La función Lambda no tiene configuración VPC

La función Lambda que Verified Permissions crea para su API autorizador no está conectada a un. VPC De forma predeterminada. APIslos que tienen acceso a la red restringido a privado no VPCs pueden comunicarse con la función Lambda que autoriza las solicitudes de acceso con permisos verificados.

Verified Permissions implementa los recursos del autorizador en CloudFormation

Para crear un almacén API de políticas vinculado, debe iniciar sesión con un AWS director con muchos privilegios en la consola de permisos verificados. Este usuario despliega una AWS CloudFormation pila que crea recursos en varios. Servicios de AWS Este director debe tener permiso para añadir y modificar recursos en Verified Permissions IAM, Lambda y API Gateway. Como práctica recomendada, no comparta estas credenciales con otros administradores de su organización.

Consulte Pasar a la producción con AWS CloudFormation para obtener una descripción general de los recursos que crea Verified Permissions.

Añadir un control de acceso basado en atributos () ABAC

Una sesión de autenticación típica con un IdP devuelve los identificadores de identificación y acceso. Puede pasarle cualquiera de estos tipos de token como token portador en las solicitudes de solicitud que le envíes. API En función de las opciones que haya elegido al crear el almacén de políticas, Verified Permissions utilizará uno de los dos tipos de token. Ambos tipos contienen información sobre la pertenencia al grupo del usuario. Para obtener más información sobre los tipos de token en Amazon Cognito, consulte Uso de tokens con grupos de usuarios en la Guía para desarrolladores de Amazon Cognito.

Tras crear un almacén de políticas, puede añadir y ampliar políticas. Por ejemplo, puede agregar nuevos grupos a sus políticas a medida que los agrega a su grupo de usuarios. Como su almacén de políticas ya conoce la forma en que su grupo de usuarios presenta los grupos en forma de tokens, puede permitir un conjunto de acciones para cualquier grupo nuevo con una política nueva.

Es posible que también desee ampliar el modelo de evaluación de políticas basado en grupos para convertirlo en un modelo más preciso basado en las propiedades de los usuarios. Los tokens del grupo de usuarios contienen información de usuario adicional que puede contribuir a las decisiones de autorización.

Tokens de identificación

> Los identificadores representan los atributos de un usuario y tienen el nivel más alto de control de acceso detallado. Para evaluar las direcciones de correo electrónico, los números de teléfono o los atributos personalizados, como el departamento y el gerente, evalúa el token de identificación.

Tokens de acceso

> Los tokens de acceso representan los permisos de un usuario con un alcance OAuth 2.0. Para añadir una capa de autorización o configurar solicitudes de recursos adicionales, evalúa el token de acceso. Por ejemplo, puede validar que un usuario esté en los grupos adecuados y tenga un ámbito como el `PetStore.read` que generalmente autoriza el acceso a losAPI. Los grupos de usuarios pueden añadir ámbitos personalizados a los tokens con servidores de recursos y con la personalización de los mismos durante el tiempo de ejecución.

Consulte, Asignación de tokens de proveedores de identidad al esquema por ejemplo, las políticas que procesan las reclamaciones en los tokens de identificación y acceso.

Pasar a la producción con AWS CloudFormation

APILos almacenes de pólizas vinculados a ellos son una forma de crear rápidamente un modelo de autorización para un API Gateway. API Están diseñados para servir como entorno de prueba para el componente de autorización de su aplicación. Después de crear su almacén de políticas de prueba, dedique tiempo a refinar las políticas, el esquema y el autorizador Lambda.

Puede ajustar su API arquitectura y requerir ajustes equivalentes en el esquema y las políticas del almacén de políticas. API-los almacenes de políticas vinculados no actualizan automáticamente su esquema desde API Architecture: Verified Permissions solo los consulta API al crear un almacén de políticas. Si API los cambios son suficientes, es posible que tengas que repetir el proceso con un nuevo almacén de políticas.

Cuando su modelo de aplicación y autorización estén listos para su implementación en producción, integre el almacén API de políticas vinculado que desarrolló con sus procesos de automatización. Como práctica recomendada, le recomendamos que exporte el esquema y las políticas del almacén de políticas a una AWS CloudFormation plantilla que pueda implementar en otros Cuentas de AWS y Regiones de AWS.

Los resultados del proceso del almacén API de políticas vinculado son un almacén de políticas inicial y un autorizador Lambda. El autorizador Lambda tiene varios recursos dependientes. Verified Permissions implementa estos recursos en una pila generada automáticamente. CloudFormation Para implementarlo en producción, debe recopilar el almacén de políticas y los recursos del autorizador Lambda en una plantilla. Un almacén API de políticas vinculado está compuesto por los siguientes recursos:

1. AWS::VerifiedPermissions:PolicyStore: Copia tu esquema en el `SchemaDefinition` objeto. Escapa de " los personajes como\".
2. AWS:::VerifiedPermissions:IdentitySource: Copie los valores de la salida de su almacén GetIdentitySource de políticas de prueba y modifíquelos según sea necesario.
3. Uno o más de los AWSsiguientes:VerifiedPermissions: :Policy: copie su declaración de política en el `Definition` objeto. Escapa de " los personajes como\".
4. AWS: :Lambda: :Function,: AWS:: :Role,IAM::: :Policy,AWSIAM::: :Authorizer,AWS ApiGateway: :Lambda: :PermisoAWS: copia la plantilla de la pestaña Plantilla de la pila que Verified Permissions implementó cuando creaste tu almacén de políticas.

La siguiente plantilla es un ejemplo de almacén de políticas. Puede añadir los recursos del autorizador Lambda de su pila existente a esta plantilla.

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "MyExamplePolicyStore": {
            "Type": "AWS::VerifiedPermissions::PolicyStore",
            "Properties": {
                "ValidationSettings": {
                    "Mode": "STRICT"
                },
                "Description": "ApiGateway: PetStore/test",
                "Schema": {
                    "CedarJson": "{\"PetStore\":{\"actions\":{\"get /pets\":
{\"appliesTo\":{\"principalTypes\":[\"User\"],\"resourceTypes\":[\"Application\"],
\"context\":{\"type\":\"Record\",\"attributes\":{}}}},\"get /\":{\"appliesTo\":
{\"principalTypes\":[\"User\"],\"resourceTypes\":[\"Application\"],\"context\":{\"type
\":\"Record\",\"attributes\":{}}}},\"get /pets/{petId}\":{\"appliesTo\":{\"context
\":{\"type\":\"Record\",\"attributes\":{}},\"resourceTypes\":[\"Application\"],
\"principalTypes\":[\"User\"]}},\"post /pets\":{\"appliesTo\":{\"principalTypes\":
[\"User\"],\"resourceTypes\":[\"Application\"],\"context\":{\"type\":\"Record\",
\"attributes\":{}}}}},\"entityTypes\":{\"Application\":{\"shape\":{\"type\":\"Record\",
\"attributes\":{}}},\"User\":{\"memberOfTypes\":[\"UserGroup\"],\"shape\":{\"attributes
\":{},\"type\":\"Record\"}},\"UserGroup\":{\"shape\":{\"type\":\"Record\",\"attributes
\":{}}}}}}"
                }
            }
        },
        "MyExamplePolicy": {
            "Type": "AWS::VerifiedPermissions::Policy",
            "Properties": {
                "Definition": {
                    "Static": {
                        "Description": "Policy defining permissions for testgroup
 cognito group",
                        "Statement": "permit(\nprincipal in PetStore::UserGroup::
\"us-east-1_EXAMPLE|testgroup\",\naction in [\n  PetStore::Action::\"get /\",
\n  PetStore::Action::\"post /pets\",\n  PetStore::Action::\"get /pets\",\n
 PetStore::Action::\"get /pets/{petId}\"\n],\nresource);"
                    }
                },
                "PolicyStoreId": {
```

```
                    "Ref": "MyExamplePolicyStore"
                }
            },
            "DependsOn": [
                "MyExamplePolicyStore"
            ]
        },
        "MyExampleIdentitySource": {
            "Type": "AWS::VerifiedPermissions::IdentitySource",
            "Properties": {
                "Configuration": {
                    "CognitoUserPoolConfiguration": {
                        "ClientIds": [
                            "1example23456789"
                        ],
                        "GroupConfiguration": {
                            "GroupEntityType": "PetStore::UserGroup"
                        },
                        "UserPoolArn": "arn:aws:cognito-idp:us-
east-1:123456789012:userpool/us-east-1_EXAMPLE"
                    }
                },
                "PolicyStoreId": {
                    "Ref": "MyExamplePolicyStore"
                },
                "PrincipalEntityType": "PetStore::User"
            },
            "DependsOn": [
                "MyExamplePolicyStore"
            ]
        }
    }
}
```

Solución de problemas de almacenes de políticas vinculados API

Usa la información aquí para ayudarte a diagnosticar y solucionar problemas comunes al crear almacenes de políticas API vinculados a permisos verificados de Amazon.

Temas

- He actualizado mi política, pero la decisión de autorización no ha cambiado
- He adjuntado el autorizador Lambda a miAPI, pero no genera solicitudes de autorización

- He recibido una decisión de autorización inesperada y quiero revisar la lógica de autorización
- Quiero buscar los registros de mi autorizador Lambda
- Mi autorizador Lambda no existe
- Mi cuenta API está en un entorno privado VPC y no puedo invocar al autorizador
- Quiero procesar atributos de usuario adicionales en mi modelo de autorización
- Quiero añadir nuevas acciones, atributos de contexto de acción o atributos de recursos

He actualizado mi política, pero la decisión de autorización no ha cambiado

De forma predeterminada, Verified Permissions configura el autorizador Lambda para almacenar en caché las decisiones de autorización durante 120 segundos. Vuelva a intentarlo transcurridos dos minutos o desactive la memoria caché de su autorizador. Para obtener más información, consulte Habilitar el almacenamiento en API caché para mejorar la capacidad de respuesta en la Guía para desarrolladores de Amazon API Gateway.

He adjuntado el autorizador Lambda a miAPI, pero no genera solicitudes de autorización

Para empezar a procesar las solicitudes, debe implementar la API etapa a la que adjuntó su autorizador. Para obtener más información, consulte Implementación de a REST API en la Guía para desarrolladores de Amazon API Gateway.

He recibido una decisión de autorización inesperada y quiero revisar la lógica de autorización

El proceso del almacén de políticas API vinculado crea una función Lambda para su autorizador. Verified Permissions incorpora automáticamente la lógica de sus decisiones de autorización a la función de autorización. Tras crear el almacén de políticas, puede volver a revisar y actualizar la lógica de la función.

Para localizar la función Lambda desde la AWS CloudFormation consola, pulse el botón Comprobar despliegue en la página de descripción general del nuevo almacén de políticas.

También puede localizar la función en la AWS Lambda consola. Navegue hasta la consola en el almacén Región de AWS de políticas y busque el nombre de una función con el prefijo de`AVPAuthorizerLambda`. Si ha creado más de un almacén API de políticas vinculado, utilice la

hora de la última modificación de sus funciones para correlacionarlas con la creación del almacén de políticas.

Quiero buscar los registros de mi autorizador Lambda

Las funciones Lambda recopilan métricas y registran sus resultados de invocación en Amazon. CloudWatch Para revisar los registros, localice la función en la consola de Lambda y seleccione la pestaña Supervisar. Seleccione Ver CloudWatch registros y revise las entradas del grupo de registros.

Para obtener más información sobre los registros de funciones de Lambda, consulte Uso de Amazon CloudWatch Logs con AWS Lambda en la Guía para AWS Lambda desarrolladores.

Mi autorizador Lambda no existe

Tras completar la configuración de un almacén API de políticas vinculado, debe adjuntar el autorizador Lambda a su. API Si no encuentra su autorizador en la consola de API Gateway, es posible que los recursos adicionales de su almacén de políticas hayan fallado o que aún no se hayan implementado. APILos almacenes de políticas enlazados despliegan estos recursos en una AWS CloudFormation pila.

Verified Permissions muestra un enlace con la etiqueta Comprobar el despliegue al final del proceso de creación. Si ya has salido de esta pantalla, ve a la CloudFormation consola y busca en las pilas recientes un nombre que lleve el prefijo. `AVPAuthorizer-<policy store ID>` CloudFormation proporciona información valiosa sobre la solución de problemas en el resultado de una implementación de stack.

Para obtener ayuda con la solución de problemas de CloudFormation pilas, consulte Solución de problemas CloudFormation en la Guía del AWS CloudFormation usuario.

Mi cuenta API está en un entorno privado VPC y no puedo invocar al autorizador

Los permisos verificados no admiten el acceso a los autorizadores de Lambda a través de puntos finales. VPC Debe abrir una ruta de red entre su función Lambda API y la que actúa como su autorizador.

Quiero procesar atributos de usuario adicionales en mi modelo de autorización

El proceso API de almacenamiento de políticas vinculado a este sistema deriva las políticas de permisos verificados de las declaraciones del grupo en los tokens de los usuarios. Para actualizar

su modelo de autorización y tener en cuenta otros atributos de usuario, integre esos atributos en sus políticas.

Puede asignar muchas reclamaciones de los tokens de ID y acceso de los grupos de usuarios de Amazon Cognito a las declaraciones de la política de permisos verificados. Por ejemplo, la mayoría de los usuarios tienen una `email` reclamación en su token de identificación. Para obtener más información sobre cómo añadir las reclamaciones de tu fuente de identidad a las políticas, consultaAsignación de tokens de proveedores de identidad al esquema.

Quiero añadir nuevas acciones, atributos de contexto de acción o atributos de recursos

Un almacén API de políticas vinculado y el autorizador Lambda que crea son un recurso. point-in-time Reflejan tu estado API en el momento de la creación. El esquema del almacén de políticas no asigna ningún atributo de contexto a las acciones, ni ningún atributo o elemento principal al `Application` recurso predeterminado.

Al añadir acciones (rutas y métodos) a la suyaAPI, debe actualizar el almacén de políticas para estar al tanto de las nuevas acciones. También debe actualizar su autorizador Lambda para procesar las solicitudes de autorización para las nuevas acciones. Puede empezar de nuevo con un almacén de políticas nuevo o actualizar el almacén de políticas existente.

Para actualizar tu almacén de políticas existente, localiza tu función. Examine la lógica de la función generada automáticamente y actualícela para procesar las nuevas acciones, atributos o contextos. A continuación, edite el esquema para incluir las nuevas acciones y atributos.

Eliminar almacenes de políticas

Puedes eliminar los almacenes de políticas de permisos verificados de Amazon mediante el AWS Management Console o el AWS CLI. Al eliminar un almacén de políticas, se elimina permanentemente el esquema y cualquier política del almacén de políticas.

Es posible que desee eliminar los almacenes de políticas por los siguientes motivos:

- Ha alcanzado la cuota de almacenes de pólizas disponibles en una región determinada. Para obtener más información, consulte Cuotas de recursos.
- Ya no apoyas a un inquilino en una solicitud con varios inquilinos y, por lo tanto, ya no necesitas ese almacén de pólizas.

AWS Management Console

Para eliminar un almacén de políticas

1. Abre la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación izquierdo, elija Configuración.
3. Seleccione Eliminar este almacén de políticas.
4. Escriba `delete` en la casilla de texto y seleccione Eliminar.

AWS CLI

Para eliminar un almacén de políticas

Puede eliminar un almacén de políticas mediante la operación `delete-policy-store`.

```
$ aws verifiedpermissions delete-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

Este comando no genera ninguna salida si se realiza correctamente.

Esquema del almacén de políticas de Amazon Verified Permissions.

Un esquema es una declaración de la estructura de los tipos de entidad que admite la aplicación y de las acciones que la aplicación puede proporcionar en las solicitudes de autorización.

Para obtener información más detallada consulte Formato de esquemas Cedar en la Guía de referencia sobre el lenguaje de las políticas de Cedar.

> **Note**
>
> El uso de esquemas en Verified Permissions es opcional, pero se recomienda encarecidamente su uso para el software de producción. Cuando se crea una nueva política, Verified Permissions puede usar el esquema para validar las entidades y los atributos a los que se hace referencia en el ámbito y las condiciones a fin de evitar errores tipográficos y errores en las políticas que puedan provocar un comportamiento errático del sistema. Si activa la validación de políticas, todas las políticas nuevas deben ajustarse al esquema.

AWS Management Console

Para crear un esquema

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, seleccione Esquema.
3. Elija Create schema (Crear esquema).

AWS CLI

Para enviar un esquema nuevo o sobrescribir un esquema existente mediante la AWS CLI.

Puede crear un almacén de políticas ejecutando un AWS CLI comando similar al siguiente ejemplo.

Considere un esquema que contenga el siguiente contenido de Cedar:

```
{
```

```
    "MySampleNamespace": {
        "actions": {
            "remoteAccess": {
                "appliesTo": {
                    "principalTypes": [ "Employee" ]
                }
            }
        },
        "entityTypes": {
            "Employee": {
                "shape": {
                    "type": "Record",
                    "attributes": {
                        "jobLevel": {"type": "Long"},
                        "name": {"type": "String"}
                    }
                }
            }
        }
    }
}
```

Primero debe escaparse JSON a una cadena de una sola línea y comenzar con una declaración de su tipo de datos:`cedarJson`. El siguiente ejemplo utiliza el siguiente contenido del `schema.json` archivo que contiene la versión escapada del JSON esquema.

> **Note**
>
> El ejemplo se muestra con ajustes de línea para facilitar la lectura. Debe tener todo el archivo en una sola línea para que el comando lo acepte.

```
{"cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {\"appliesTo
\":
{\"principalTypes\": [\"Employee\"]}}},\"entityTypes\": {\"Employee\": {\"shape\":
{\"attributes\": {\"jobLevel\": {\"type\": \"Long\"},\"name\": {\"type\": \"String
\"}},
\"type\": \"Record\"}}}}}"}
```

```
$ aws verifiedpermissions put-schema \
    --definition file://schema.json \
```

```
    --policy-store PSEXAMPLEabcdefg111111
{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "namespaces": [
        "MySampleNamespace"
    ],
    "createdDate": "2023-07-17T21:07:43.659196+00:00",
    "lastUpdatedDate": "2023-08-16T17:03:53.081839+00:00"
}
```

AWS SDKs

Puede crear un almacén de políticas mediante `PutSchema`API. Para obtener más información, consulta PutSchemala Guía de API referencia de permisos verificados de Amazon.

Edición de esquemas de almacenes de políticas en modo visual

Al seleccionar Schema en la consola de permisos verificados de Amazon, el modo visual muestra los tipos de entidad y las acciones que componen el esquema. En esta vista de nivel superior o desde los detalles de cualquier entidad, puede elegir Editar esquema para empezar a realizar actualizaciones en su esquema. El modo visual no está disponible con algunos formatos de esquema, como los registros anidados.

El editor visual de esquemas comienza con una serie de diagramas que ilustran las relaciones entre las entidades del esquema. Elija Expandir para maximizar la vista de los diagramas. Hay dos diagramas disponibles:

- Diagrama de acciones: la vista del diagrama de acciones muestra los tipos de directores que ha configurado en su almacén de políticas, las acciones que pueden realizar y los recursos sobre los que pueden realizar acciones. Las líneas entre las entidades indican su capacidad para crear una política que permita a un director realizar una acción sobre un recurso. Si el diagrama de acciones no indica una relación entre dos entidades, debe crear esa relación entre ellas para poder permitirla o denegarla en las políticas. Seleccione una entidad para ver un resumen de las propiedades y profundice para ver todos los detalles. Seleccione Filtrar por [acción | tipo de recurso | tipo principal] para ver una entidad en una vista con solo sus propias conexiones.
- Diagrama de tipos de entidades: el diagrama de tipos de entidades se centra en las relaciones entre los principales y los recursos. Cuando desee comprender las complejas relaciones principales anidadas en su esquema, revise este diagrama. Pase el ratón sobre una entidad para profundizar en las relaciones principales que tiene.

Debajo de los diagramas hay vistas de lista de los tipos de entidades y las acciones del esquema. La vista de lista resulta útil cuando se desean ver inmediatamente los detalles de una acción o un tipo de entidad específicos. Seleccione cualquier entidad para ver los detalles.

Para editar un esquema de Verified Permissions en modo visual

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, seleccione Esquema.
3. Elija Modo visual. Revise los diagramas entidad-relación y planifique los cambios que desee realizar en el esquema. Si lo desea, puede filtrar por una entidad para examinar sus conexiones individuales con otras entidades.
4. Elija Edit schema (Editar esquema).
5. En la sección Detalles, escriba un espacio de nombres para su esquema.
6. En la sección Tipos de entidad, seleccione Agregar nuevo tipo de entidad.
7. Escriba el nombre de la entidad.
8. (Opcional) Seleccione Agregar un elemento principal para añadir las entidades principales a las que pertenece la nueva entidad. Para eliminar un elemento principal que se haya agregado a la entidad, seleccione Eliminar junto al nombre del elemento principal.
9. Para añadir otro atributo, seleccione Agregar un atributo. Escriba el nombre del atributo y elija el tipo de atributo para cada atributo de la entidad. Verified Permissions utiliza los valores de atributo especificados al verificar las políticas con el esquema. Seleccione si cada atributo es obligatorio. Para eliminar un atributo que se ha añadido a la entidad, seleccione Eliminar junto al atributo.
10. Seleccione Agregar tipo de entidad para añadir la entidad al esquema.
11. En la sección Acciones, seleccione Agregar acción nueva.
12. Escriba el nombre de la acción.
13. (Opcional) Seleccione Agregar un recurso para añadir los tipos de recurso a los que se refiere la acción. Para eliminar un tipo de recurso que se ha agregado a la acción, seleccione Eliminar junto al nombre del tipo de recurso.
14. (Opcional) Seleccione Agregar una entidad principal para añadir el tipo de entidad principal al que se refiere la acción. Para eliminar un tipo de entidad principal que se ha agregado a la acción, seleccione Eliminar junto al nombre del tipo de entidad principal.

15. Seleccione Añadir un atributo para añadir atributos que se puedan añadir al contexto de una acción en sus solicitudes de autorización. Introduzca el nombre del atributo y elija el tipo de atributo para cada atributo. Verified Permissions utiliza los valores de atributo especificados al verificar las políticas con el esquema. Seleccione si cada atributo es obligatorio. Para eliminar un atributo que se ha añadido a la acción, seleccione Eliminar junto al atributo.
16. Seleccione Agregar acción.
17. Una vez que se hayan agregado todos los tipos de entidad y acciones al esquema, elija Guardar cambios.

Edición de esquemas de almacenes de políticas en modo JSON

Al seleccionar Schema en la consola de permisos verificados de Amazon, el JSONmodo muestra los tipos de entidad y las acciones que componen el esquema. Al elegir Editar esquema, puede empezar a actualizar el JSON código del esquema directamente en el JSON editor. Mientras realizas las actualizaciones, te darás cuenta de que el JSON editor valida el código según la JSON sintaxis e identificará los errores y las advertencias a medida que lo edites, lo que te permitirá encontrar los problemas rápidamente. Además, no tienes que preocuparte por el formato delJSON. Solo tienes que elegir Formato JSON una vez que hayas realizado las actualizaciones y el formato se actualizará para que coincida con el JSON formato esperado.

Para editar un esquema de permisos verificados en el JSON modo

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, seleccione Esquema.
3. Elija JSONel modo y, a continuación, elija Editar esquema.
4. Introduce el contenido del JSON esquema en el campo Contenido. No puede guardar las actualizaciones de su esquema hasta que resuelva todos los errores de sintaxis. Puede elegir Formato JSON para formatear la JSON sintaxis del esquema con el espaciado y la sangría recomendados.
5. Elija Guardar cambios.

Activación del modo de validación de la política de permisos verificados de Amazon

Puede configurar el modo de validación de políticas en Verified Permissions para controlar si los cambios en las políticas se validan con respecto al esquema de su almacén de políticas.

> **Important**
>
> Al activar la validación de políticas, todos los intentos de crear o actualizar una política o una plantilla de política se validan con el esquema del almacén de políticas. Verified Permissions rechaza la solicitud si se produce un error en la validación.

AWS Management Console

Para configurar el modo de validación de políticas de un almacén de políticas

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. Elija Configuración.
3. En la sección Modo de validación de políticas, seleccione Modificar.
4. Realice una de las siguientes acciones siguientes:
   - Para activar la validación de políticas y hacer que todos los cambios en las políticas se validen según su esquema, pulse el botón de opción Estricto (recomendado).
   - Para desactivar la validación de políticas en caso de cambios en las políticas, pulse el botón de opción Desactivada. Escriba `confirm` para confirmar que las actualizaciones de las políticas ya no se validarán según su esquema.
5. Elija Guardar cambios.

AWS CLI

Para configurar el modo de validación de un almacén de políticas

Puede cambiar el modo de validación de un almacén de políticas mediante la UpdatePolicyStoreoperación y especificando un valor diferente para el ValidationSettingsparámetro.

```
$ aws verifiedpermissions update-policy-store \
    --validation-settings "mode=OFF",
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-17T18:36:10.134448+00:00",
    "lastUpdatedDate": "2023-05-17T18:36:10.134448+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "validationSettings": {
        "Mode": "OFF"
    }
}
```

Para obtener información más detallada consulte Validación de políticas en la Guía de referencia sobre el lenguaje de las políticas de Cedar.

Políticas de Amazon Verified Permissions

Una política es una instrucción que permite o prohíbe a una entidad principal realizar una o más acciones en un recurso. Cada política se evalúa de forma independiente respecto de cualquier otra política. Para obtener más información sobre cómo se estructuran y evalúan las políticas de Cedar, consulte la sección sobre la validación de las políticas de Cedar con el esquema en la Guía de referencia sobre el lenguaje de las políticas de Cedar.

Important

Cuando redacte las políticas de Cedar que hagan referencia a las entidades principales, los recursos y las acciones, puede definir los identificadores únicos que se utilizan para cada uno de esos elementos. Le recomendamos encarecidamente que siga las siguientes prácticas recomendadas:

- Utilice valores como los identificadores únicos universales (UUIDs) para todos los identificadores principales y de recursos.

  Por ejemplo, si el usuario `jane` deja la empresa y luego permite que otra persona use el nombre `jane`, ese nuevo usuario tendrá acceso automáticamente a todo lo que otorgan las políticas que aún hacen referencia a `User::"jane"`. Cedar no puede distinguir entre el usuario nuevo y el antiguo. Esto también se aplica a los identificadores de las entidades principales y de los recursos. Utilice siempre identificadores con garantías de que son únicos y que no se han reutilizado nunca para asegurarse de no conceder acceso involuntariamente debido a la presencia de un identificador antiguo en una política.

  Si utilizas a UUID para una entidad, te recomendamos que sigas el especificador de comentarios//y el nombre descriptivo de la entidad. Esto ayuda a que sus políticas sean más fáciles de entender. Por ejemplo: principal == User: :"a1b2c3d4-e5f6-a1b2-c3d4- «,// alice EXAMPLE11111
- No incluya información de identificación personal, confidencial o sensible como parte del identificador único de sus entidades principales o recursos. Estos identificadores se incluyen en las entradas de registro que se comparten en las rutas. AWS CloudTrail

Temas

- Creación de políticas estáticas de Amazon Verified Permissions
- Edición de políticas estáticas de Amazon Verified Permissions
- Uso del banco de pruebas de permisos verificados de Amazon
- Ejemplo de políticas de Amazon Verified Permissions

Formato de entidades en las políticas de Amazon Verified Permissions

Amazon Verified Permissions utiliza el lenguaje de las políticas de Cedar para crear políticas. La sintaxis de las políticas y los tipos de datos admitidos coinciden con la sintaxis y los tipos de datos descritos en los temas de Construcción básica de políticas en Cedar y Tipos de datos compatibles con Cedar de la Guía de referencia sobre el lenguaje de políticas de Cedar. Sin embargo, existen diferencias entre Verified Permissions y Cedar en cuanto al formato de las entidades cuando se realiza una solicitud de autorización.

El JSON formato de las entidades en Verified Permissions difiere del de Cedar en los siguientes aspectos:

- En los permisos verificados, un JSON objeto debe tener todos sus pares clave-valor incluidos en un JSON objeto con el nombre de. `Record`
- Una JSON lista de permisos verificados debe estar agrupada en un par JSON clave-valor en el que el nombre de la clave sea `Set` y el valor sea la lista original JSON de Cedar.
- `String`En el caso `Long` de los nombres `Boolean` de tipo y de tipo, cada par clave-valor de Cedar se sustituye por un JSON objeto en Verified Permissions. El nombre del objeto es el nombre de la clave original. Dentro del JSON objeto, hay un par clave-valor en el que el nombre de la clave es el nombre de tipo del valor escalar (`String`,`Long`, o`Boolean`) y el valor es el valor de la entidad Cedar.
- El formato de la sintaxis de las entidades de Cedar y Verified Permissions difiere en los siguientes aspectos:

Formato de Cedar	Formato de Verified Permissions
`uid`	`Identifier`
`type`	`EntityType`

Formato de Cedar	Formato de Verified Permissions
`id`	`EntityId`
`attrs`	`Attributes`
`parents`	`Parents`

Example - Listas

Los siguientes ejemplos muestran cómo se expresa una lista de entidades en Cedar y Verified Permissions, respectivamente.

Cedar

```
[
  {
    "number": 1
  },
  {
    "sentence": "Here is an example sentence"
  },
  {
    "Question": false
  }
]
```

Verified Permissions

```
{
  "Set": [
    {
      "Record": {
        "number": {
          "Long": 1
        }
      }
    },
    {
      "Record": {
        "sentence": {
```

```
          "String": "Here is an example sentence"
        }
      }
    },
    {
      "Record": {
        "question": {
          "Boolean": false
        }
      }
    }
  ]
}
```

Example - Evaluación de políticas

Los siguientes ejemplos muestran cómo se formatean las entidades para evaluar una política en una solicitud de autorización en Cedar y Verified Permissions, respectivamente.

Cedar

```
[
    {
        "uid": {
            "type": "PhotoApp::User",
            "id": "alice"
        },
        "attrs": {
            "age": 25,
            "name": "alice",
            "userId": "123456789012"
        },
        "parents": [
            {
                "type": "PhotoApp::UserGroup",
                "id": "alice_friends"
            },
            {
                "type": "PhotoApp::UserGroup",
                "id": "AVTeam"
            }
        ]
```

```
    },
    {
        "uid": {
            "type": "PhotoApp::Photo",
            "id": "vacationPhoto.jpg"
        },
        "attrs": {
            "private": false,
            "account": {
                "__entity": {
                    "type": "PhotoApp::Account",
                    "id": "ahmad"
                }
            }
        },
        "parents": []
    },
    {
        "uid": {
            "type": "PhotoApp::UserGroup",
            "id": "alice_friends"
        },
        "attrs": {},
        "parents": []
    },
    {
        "uid": {
            "type": "PhotoApp::UserGroup",
            "id": "AVTeam"
        },
        "attrs": {},
        "parents": []
    }
]
```

Verified Permissions

```
[
    {
        "Identifier": {
            "EntityType": "PhotoApp::User",
            "EntityId": "alice"
        },
```

```
        "Attributes": {
            "age": {
                "Long": 25
            },
            "name": {
                "String": "alice"
            },
            "userId": {
                "String": "123456789012"
            }
        },
        "Parents": [
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "alice_friends"
            },
            {
                "EntityType": "PhotoApp::UserGroup",
                "EntityId": "AVTeam"
            }
        ]
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::Photo",
            "EntityId": "vacationPhoto.jpg"
        },
        "Attributes": {
            "private": {
                "Boolean": false
            },
            "account": {
                "EntityIdentifier": {
                    "EntityType": "PhotoApp::Account",
                    "EntityId": "ahmad"
                }
            }
        },
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "alice_friends"
```

```
        },
        "Parents": []
    },
    {
        "Identifier": {
            "EntityType": "PhotoApp::UserGroup",
            "EntityId": "AVTeam"
        },
        "Parents": []
    }
]
```

Creación de políticas estáticas de Amazon Verified Permissions

Puede crear una política estática para que los directores les permitan o prohíban realizar acciones específicas en recursos específicos para su aplicación.

AWS Management Console

Para crear una política estática

1. Abra la consola de permisos verificados en. https://console.aws.amazon.com/verifiedpermissions/ Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Políticas.
3. Seleccione Crear política y, a continuación, elija Crear política estática.
4. En la sección Efecto de la política, seleccione si la política permitirá o prohibirá una acción cuando una solicitud coincida con la política.
5. En el campo Ámbito de las entidades principales, elija el ámbito de las entidades principales al que se aplicará la política.

   - Elija Entidad principal específica para aplicar la política a una entidad principal concreta. Especifique el tipo de entidad y el identificador de la entidad principal a los que se permitirá o prohibirá realizar las acciones especificadas en la política.
   - Seleccione Grupo de entidades principales para aplicar la política a un grupo de entidades principales. Escriba el nombre del grupo de entidades principales en el campo Grupo de entidades principales.
   - Seleccione Todas las entidades principales para aplicar la política a todas las entidades principales de su almacén de políticas.

6. En el campo Ámbito de los recursos, elija el ámbito de los recursos al que se aplicará la política.

   - Seleccione Recursos específicos para aplicar la política a un recurso específico. Especifique el tipo de entidad y el identificador del recurso al que se debe aplicar la política.
   - Seleccione Grupo de recursos para aplicar la política a un grupo de recursos. Escriba el nombre del grupo de recursos en el campo Grupo de recursos.
   - Seleccione Todos los recursos para aplicar la política a todos los recursos de su almacén de políticas.

7. En el campo Ámbito de las acciones, elija el ámbito de los recursos al que se aplicará la política.

   - Seleccione Conjunto específico de acciones para aplicar la política a un conjunto de acciones. Seleccione las casillas de verificación situadas junto a las acciones para aplicar la política.
   - Seleccione Todas las acciones para aplicar la política a todas las acciones de su almacén de políticas.

8. Elija Next (Siguiente).
9. En la sección Política, revise su política de Cedar. Puede elegir Formato para dar formato a la sintaxis de su política con el espaciado y la sangría recomendados. Para obtener información más detallada consulte el tema sobre la construcción básica de políticas en Cedar en la Guía de referencia sobre el lenguaje de las políticas de Cedar.
10. En la sección Detalles, escriba una descripción opcional de la política.
11. Elija Crear política.

AWS CLI

Para crear una política estática

Puede crear una política estática mediante la CreatePolicyoperación. En el ejemplo siguiente se crea una política estática sencilla.

```
$ aws verifiedpermissions create-policy \
    --definition "{ \"static\": { \"Description\": \"MyTestPolicy\", \"Statement\":
 \"permit(principal,action,resource) when {principal.owner == resource.owner};\"}}"
 \
```

```
    --policy-store-id PSEXAMPLEabcdefg111111
{
"Arn": "arn:aws:verifiedpermissions::123456789012:policy/PSEXAMPLEabcdefg111111/
SPEXAMPLEabcdefg111111",
    "createdDate": "2023-05-16T20:33:01.730817+00:00",
    "lastUpdatedDate": "2023-05-16T20:33:01.730817+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC"
}
```

Edición de políticas estáticas de Amazon Verified Permissions

Puede editar una política estática existente en su almacén de políticas. Solo puede actualizar directamente las políticas estáticas. Para cambiar una política vinculada a una plantilla, debe actualizar la plantilla de política. Para obtener más información, consulte Edición de plantillas de política de permisos verificados de Amazon.

Puede cambiar los siguientes elementos de una política estática:

- La `action` referenciada por la política.
- Una cláusula de condición, como `when` y `unless`.

No puede cambiar los siguientes elementos de una política estática:

- Cambio de una política de estática a vinculada a una plantilla.
- Cambio del efecto de una política estática de `permit` o `forbid`.
- La `principal` a la que hace referencia una política estática.
- El `resource` al que hace referencia una política estática.

AWS Management Console

Para editar una política estática

1. Abre la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, seleccione Políticas.

3. Seleccione el botón de opción situado junto a la política estática que desee editar y, a continuación, Editar.
4. En la sección Cuerpo de la política, actualice `action` o la cláusula de condición de su política estática. No puede actualizar el efecto de la política, `principal`, o `resource` de la política.
5. Elija Actualizar política.

> **Note**
>
> Si la validación de políticas está habilitada en el almacén de políticas, la actualización de una política estática hace que Verified Permissions valide la política con el esquema del almacén de políticas. Si la política estática actualizada no supera la validación, se produce un error en la operación y la actualización no se guarda.

AWS CLI

Para editar una política estática

Puede editar una política estática mediante la UpdatePolicyoperación. En el ejemplo siguiente se edita una política estática sencilla.

En el ejemplo, se utiliza el archivo `definition.txt` para incluir la definición de la política.

```
{
    "static": {
        "description":  "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
    }
}
```

El siguiente comando hace referencia a ese archivo.

```
$ aws verifiedpermissions create-policy \
    --definition file://definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

```
{
    "createdDate": "2023-06-12T20:33:37.382907+00:00",
    "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
    },
    "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
    }
}
```

Uso del banco de pruebas de permisos verificados de Amazon

Utilice el banco de pruebas de permisos verificados para probar las políticas de permisos verificados y solucionar sus problemas mediante la ejecución de solicitudes de autorización en función de ellas. El banco de pruebas utiliza los parámetros que usted especifique para determinar si las políticas de Cedar de su almacén de políticas autorizarían la solicitud. Puede cambiar entre el modo visual y el JSONmodo mientras prueba las solicitudes de autorización. Para obtener más información sobre cómo se estructuran y evalúan las políticas de Cedar, consulte la sección sobre la construcción básica de políticas en Cedar en la Guía de referencia sobre el lenguaje de las políticas de Cedar.

> **Note**
>
> Al realizar una solicitud de autorización mediante Verified Permissions, puede proporcionar la lista de entidades principales y recursos como parte de la solicitud en la sección Entidades adicionales. Sin embargo, no es posible incluir los detalles de las acciones. Deben especificarse en el esquema o deducirse de la solicitud. No puede incluir una acción en la sección Entidades adicionales.

Para obtener una descripción visual y una demostración del banco de pruebas, consulte Amazon Verified Permissions: Policy Creation and Testing (Primer Series #3) en el AWS YouTube canal.

Visual mode

> **Note**
>
> Debe tener un esquema definido en su almacén de políticas para utilizar el modo visual del banco de pruebas.

Para probar las políticas en modo visual

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, seleccione Banco de pruebas.
3. Elija Modo visual.
4. En la sección Entidad principal, elija Entidad principal que realiza la acción entre los tipos de entidad principal del esquema. Escriba un identificador para la entidad principal en el cuadro de texto.
5. (Opcional) Seleccione Agregar un elemento principal para añadir las entidades principales del elemento principal especificado. Para eliminar un elemento principal que se haya agregado a la entidad principal, seleccione Eliminar junto al nombre del elemento principal.
6. Especifique el valor de atributo para cada atributo de la entidad principal especificada. El banco de pruebas utiliza los valores de atributo especificados en la solicitud de autorización simulada.
7. En la sección Recursos, elija el recurso sobre el que actúa la entidad principal. Escriba un identificador para el recurso en el cuadro de texto.
8. (Opcional) Seleccione Agregar un elemento principal para añadir las entidades principales del recurso especificado. Para eliminar un elemento principal que se haya agregado al recurso, seleccione Eliminar junto al nombre del elemento principal.
9. Especifique el valor de atributo para cada atributo del recurso especificado. El banco de pruebas utiliza los valores de atributo especificados en la solicitud de autorización simulada.
10. En la sección Acción, elija Acción que va a tomar la entidad principal en la lista de acciones válidas para la entidad principal y el recurso especificados.
11. Especifique el valor de atributo para cada atributo de la acción especificada. El banco de pruebas utiliza los valores de atributo especificados en la solicitud de autorización simulada.

12. (Opcional) En la sección Entidades adicionales, elija Agregar entidad para agregar las entidades que se evaluarán para la decisión de autorización.
13. Elija el identificador de entidad en la lista desplegable y escriba el identificador de entidad.
14. (Opcional) Seleccione Agregar un elemento principal para añadir las entidades principales de la entidad especificada. Para eliminar un elemento principal que se haya agregado a la entidad, seleccione Eliminar junto al nombre del elemento principal.
15. Especifique el valor de atributo para cada atributo de la entidad especificada. El banco de pruebas utiliza los valores de atributo especificados en la solicitud de autorización simulada.
16. Elija Confirmar para añadir la entidad al banco de pruebas.
17. Seleccione Ejecutar solicitud de autorización para simular la solicitud de autorización para las políticas de Cedar en el almacén de políticas. El banco de pruebas muestra la decisión de aceptar o denegar la solicitud junto con información sobre las políticas satisfechas o los errores encontrados durante la evaluación.

JSON mode

Para probar las políticas en JSON modo

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, seleccione Banco de pruebas.
3. Elige JSONel modo.
4. En la sección Detalles de la solicitud, si tiene un esquema definido, elija Entidad principal que realiza la acción entre los tipos de entidad principal de su esquema. Escriba un identificador para la entidad principal en el cuadro de texto.

   Si no tiene un esquema definido, escriba la entidad principal en el cuadro de texto Entidad principal que realiza la acción.
5. Si tiene un esquema definido, elija el recurso entre los tipos de recurso del esquema. Escriba un identificador para el recurso en el cuadro de texto.

   Si no tiene un esquema definido, escriba el recurso en el cuadro de texto Recurso.
6. Si tiene un esquema definido, elija la acción en la lista de acciones válidas para la entidad principal y el recurso especificados.

   Si no tiene un esquema definido, escriba el recurso en el cuadro de texto Acción.

7. Introduzca el contexto de la solicitud de simulación en el campo Contexto. El contexto de la solicitud es información adicional que se puede utilizar para tomar decisiones de autorización.
8. En el campo Entidades, introduzca la jerarquía de las entidades y los atributos que se van a evaluar para la decisión de autorización.
9. Seleccione Ejecutar solicitud de autorización para simular la solicitud de autorización para las políticas de Cedar en el almacén de políticas. El banco de pruebas muestra la decisión de aceptar o denegar la solicitud junto con información sobre las políticas satisfechas o los errores encontrados durante la evaluación.

Ejemplo de políticas de Amazon Verified Permissions

Los siguientes ejemplos de políticas de permisos verificados se basan en el esquema definido para la aplicación hipotética denominada PhotoFlash descrita en la sección de ejemplos de esquemas de la Guía de referencia del lenguaje de políticas de Cedar. Para obtener más información sobre la sintaxis de las políticas de Cedar, consulte el tema sobre la construcción básica de políticas en Cedar en la Guía de referencia sobre el lenguaje de las políticas de Cedar.

Ejemplos de políticas

- Permite el acceso a entidades individuales
- Permite el acceso a grupos de entidades
- Permite el acceso a cualquier entidad
- Permite el acceso a los atributos de una entidad (ABAC)
- Acceso denegado
- Utiliza la notación entre corchetes para hacer referencia a los atributos del token
- Utiliza la notación de puntos para hacer referencia a los atributos
- Refleja los atributos del token de Amazon Cognito ID
- Refleja los atributos OIDC del token de ID
- Refleja los atributos del token de acceso de Amazon Cognito
- Refleja OIDC los atributos del token de acceso

Permite el acceso a entidades individuales

En el siguiente ejemplo, se muestra cómo se puede crear una política que permita `alice` al usuario ver la foto`VacationPhoto94.jpg`.

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
);
```

Permite el acceso a grupos de entidades

En el siguiente ejemplo se muestra cómo se puede crear una política que permita `alice_friends` a cualquier persona del grupo ver la foto`VacationPhoto94.jpg`.

```
permit(
  principal in Group::"alice_friends",
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
);
```

En el siguiente ejemplo se muestra cómo se puede crear una política que permita `alice` al usuario ver cualquier foto del álbum`alice_vacation`.

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource in Album::"alice_vacation"
);
```

En el siguiente ejemplo se muestra cómo se puede crear una política que permita `alice` al usuario ver, editar o eliminar cualquier foto del álbum`alice_vacation`.

```
permit(
  principal == User::"alice",
  action in [Action::"view", Action::"edit", Action::"delete"],
  resource in Album::"alice_vacation"
);
```

En el siguiente ejemplo, se muestra cómo se puede crear una política que conceda permisos al usuario `alice` en el álbum`alice_vacation`, donde `admin` hay un grupo definido en la jerarquía del esquema que contiene los permisos para ver, editar y eliminar una foto.

```
permit(
  principal == User::"alice",
  action in PhotoflashRole::"admin",
  resource in Album::"alice_vacation"
);
```

En el siguiente ejemplo, se muestra cómo se puede crear una política que conceda permisos al usuario `alice` del álbum`alice_vacation`, `viewer` es decir, un grupo definido en la jerarquía del esquema que contiene el permiso para ver y comentar una foto. Al usuario `alice` también se le concede el permiso `edit` mediante la segunda acción que se indica en la política.

```
permit(
  principal == User::"alice",
  action in [PhotoflashRole::"viewer", Action::"edit"],
  resource in Album::"alice_vacation"
)
```

Permite el acceso a cualquier entidad

En el siguiente ejemplo, se muestra cómo se puede crear una política que permita a cualquier director autenticado ver el álbum`alice_vacation`.

```
permit(
  principal,
  action == Action::"view",
  resource in Album::"alice_vacation"
);
```

En el siguiente ejemplo se muestra cómo se puede crear una política que permita al usuario `alice` enumerar todos los álbumes de la `jane` cuenta, enumerar las fotos de cada álbum y ver las fotos de la cuenta.

```
permit(
  principal == User::"alice",
  action in [Action::"listAlbums", Action::"listPhotos", Action::"view"],
```

```
  resource in Account::"jane"
);
```

En el siguiente ejemplo, se muestra cómo se puede crear una política que permita `alice` al usuario realizar cualquier acción en los recursos del álbum`jane_vaction`.

```
permit(
  principal == User::"alice",
  action,
  resource in Album::"jane_vacation"
);
```

Permite el acceso a los atributos de una entidad (ABAC)

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. Verified Permissions permite adjuntar atributos a las entidades principales, las acciones y los recursos. Después se puede hacer referencia a estos atributos en las cláusulas `when` y `unless` de las políticas que evalúan los atributos de las entidades principales, las acciones y los recursos que componen el contexto de la solicitud.

Los siguientes ejemplos utilizan los atributos definidos en la aplicación hipotética denominada PhotoFlash descrita en la sección de ejemplos de esquemas de la Guía de referencia sobre el lenguaje normativo de Cedar.

En el siguiente ejemplo se muestra cómo se puede crear una política que permita a cualquier director del `HardwareEngineering` departamento con un nivel de trabajo superior o igual a 5 ver y enumerar las fotos del álbum`device_prototypes`.

```
permit(
  principal,
  action in [Action::"listPhotos", Action::"view"],
  resource in Album::"device_prototypes"
)
when {
  principal.department == "HardwareEngineering" &&
  principal.jobLevel >= 5
};
```

En el siguiente ejemplo, se muestra cómo se puede crear una política que permita `alice` al usuario ver cualquier recurso de tipo archivo`JPEG`.

```
permit(
  principal == User::"alice",
  action == Action::"view",
  resource
)
when {
  resource.fileType == "JPEG"
};
```

Las acciones tienen atributos de contexto. Debe incluir estos atributos en una solicitud `context` de autorización. En el siguiente ejemplo, se muestra cómo se puede crear una política que permita `alice` al usuario realizar cualquier `readOnly` acción. También puede establecer una `appliesTo` propiedad para las acciones del esquema. Esto especifica las acciones válidas para un recurso cuando se quiere garantizar que, por ejemplo, los usuarios solo puedan intentar autorizar `ViewPhoto` un recurso de ese tipo`PhotoFlash::Photo`.

```
permit(
  principal == PhotoFlash::User::"alice",
  action,
  resource
) when {
    context has readOnly &&
    context.readOnly == true
};
```

Sin embargo, una forma mejor de establecer las propiedades de las acciones del esquema es organizarlas en grupos de acciones funcionales. Por ejemplo, puede crear una acción con el nombre de un grupo de acciones `ReadOnlyPhotoAccess` y `PhotoFlash::Action::"ViewPhoto"` configurarla `ReadOnlyPhotoAccess` como miembro. En el siguiente ejemplo, se muestra cómo se puede crear una política que conceda a Alice acceso a las acciones de solo lectura de ese grupo.

```
permit(
  principal == PhotoFlash::User::"alice",
  action,
  resource
) when {
    action in PhotoFlash::Action::"ReadOnlyPhotoAccess"
};
```

En el siguiente ejemplo se muestra cómo se puede crear una política que permita a todos los directores realizar cualquier acción en los recursos para los que tienen atributos. `owner`

```
permit(
  principal,
  action,
  resource
)
when {
  principal == resource.owner
};
```

En el siguiente ejemplo, se muestra cómo se puede crear una política que permita a cualquier principal ver cualquier recurso si el `department` atributo del principal coincide con el `department` atributo del recurso.

> **Note**
>
> Si una entidad no tiene un atributo mencionado en una condición de la política, la política se ignorará al tomar una decisión de autorización y la evaluación de esa política fallará para esa entidad. Por ejemplo, esta política no puede conceder acceso a ningún recurso a ninguna entidad principal que no tenga un atributo `department`.

```
permit(
  principal,
  action == Action::"view",
  resource
)
when {
  principal.department == resource.owner.department
};
```

En el siguiente ejemplo, se muestra cómo se puede crear una política que permita a cualquier principal realizar cualquier acción en un recurso si el principal es el `owner` del recurso O si el principal forma parte del `admins` grupo del recurso.

```
permit(
  principal,
  action,
```

```
  resource,
)
when {
  principal == resource.owner ||
  resource.admins.contains(principal)
};
```

Acceso denegado

Si una política contiene `forbid` para el efecto de la política, restringe los permisos en lugar de concederlos.

> **Important**
>
> Durante la autorización, si se aplican tanto una política `permit` como `forbid`, la de `forbid` tiene prioridad.

En los ejemplos siguientes se utilizan los atributos definidos en la aplicación hipotética denominada PhotoFlash descrita en la sección de ejemplos de esquemas de la Guía de referencia sobre el lenguaje de políticas de Cedar.

En el siguiente ejemplo, se muestra cómo se puede crear una política que impida al usuario `alice` realizar todas las acciones excepto `readOnly` en cualquier recurso.

```
forbid (
  principal == User::"alice",
  action,
  resource
)
unless {
  action.readOnly
};
```

En el siguiente ejemplo se muestra cómo se puede crear una política que deniegue el acceso a todos los recursos que tengan un `private` atributo, a menos que el principal tenga el `owner` atributo del recurso.

```
forbid (
  principal,
```

```
  action,
  resource
)
when {
  resource.private
}
unless {
  principal == resource.owner
};
```

Utiliza la notación entre corchetes para hacer referencia a los atributos del token

En el siguiente ejemplo, se muestra cómo se puede crear una política que utilice la notación entre corchetes para hacer referencia a los atributos del token.

Para obtener más información sobre el uso de atributos de token en las políticas de Permisos verificados, consulte Asignación de tokens de proveedores de identidad al esquema

```
permit (
    principal in MyCorp::UserGroup::"us-west-2_EXAMPLE|MyUserGroup",
    action,
    resource
) when {
    principal["cognito:username"] == "alice" &&
    principal["custom:employmentStoreCode"] == "petstore-dallas" &&
    principal has email && principal.email == "alice@example.com" &&
    context["ip-address"] like "192.0.2.*"
};
```

Utiliza la notación de puntos para hacer referencia a los atributos

En el siguiente ejemplo, se muestra cómo se puede crear una política que utilice la notación de puntos para hacer referencia a los atributos.

Para obtener más información sobre el uso de atributos de token en las políticas de permisos verificados, consulte Asignación de tokens de proveedores de identidad al esquema

```
permit(principal, action, resource)
when {
```

```
    principal.cognito.username == "alice" &&
    principal.custom.employmentStoreCode == "petstore-dallas" &&
    principal.tenant == "x11app-tenant-1" &&
    principal has email && principal.email == "alice@example.com"
};
```

Refleja los atributos del token de Amazon Cognito ID

En el siguiente ejemplo, se muestra cómo puede crear una política que haga referencia a los atributos del token de ID desde Amazon Cognito.

Para obtener más información sobre el uso de atributos de token en las políticas de permisos verificados, consulte Asignación de tokens de proveedores de identidad al esquema

```
permit (
    principal in MyCorp::UserGroup::"us-west-2_EXAMPLE|MyUserGroup",
    action,
    resource
) when {
    principal["cognito:username"] == "alice" &&
    principal["custom:employmentStoreCode"] == "petstore-dallas" &&
    principal.tenant == "x11app-tenant-1" &&
    principal has email && principal.email == "alice@example.com"
};
```

Refleja los atributos OIDC del token de ID

En el siguiente ejemplo, se muestra cómo se puede crear una política que haga referencia a los atributos del token de ID de un OIDC proveedor.

Para obtener más información sobre el uso de atributos de token en las políticas de Permisos verificados, consulte Asignación de tokens de proveedores de identidad al esquema

```
permit (
    principal in MyCorp::UserGroup::"MyOIDCProvider|MyUserGroup",
    action,
    resource
) when {
    principal.email_verified == true && principal.email == "alice@example.com" &&
    principal.phone_number_verified == true && principal.phone_number like "+1206*"
};
```

Refleja los atributos del token de acceso de Amazon Cognito

En el siguiente ejemplo, se muestra cómo puede crear una política que haga referencia a los atributos del token de acceso desde Amazon Cognito.

Para obtener más información sobre el uso de atributos de token en las políticas de permisos verificados, consulte Asignación de tokens de proveedores de identidad al esquema

```
permit(principal, action in [MyApplication::Action::"Read",
 MyApplication::Action::"GetStoreInventory"], resource)
when {
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
    context.token.scope.contains("MyAPI/mydata.write")
};
```

Refleja OIDC los atributos del token de acceso

En el siguiente ejemplo, se muestra cómo se puede crear una política que haga referencia a los atributos del token de acceso de un OIDC proveedor.

Para obtener más información sobre el uso de atributos de token en las políticas de Permisos verificados, consulte Asignación de tokens de proveedores de identidad al esquema

```
permit(
    principal,
    action in [MyApplication::Action::"Read",
 MyApplication::Action::"GetStoreInventory"],
    resource
)
when {
    context.token.client_id == "52n97d5afhfiu1c4di1k5m8f60" &&
    context.token.scope.contains("MyAPI-read")
};
```

Plantillas de políticas de permisos verificados de Amazon y políticas vinculadas a plantillas

En los permisos verificados, las plantillas de políticas son políticas con marcadores de posición para los permisos verificados `principal` `resource`, o para ambos. Las plantillas de políticas por sí solas no se pueden utilizar para gestionar las solicitudes de autorización. Para gestionar las solicitudes de autorización, se debe crear una política vinculada a una plantilla basada en una plantilla de política. Las plantillas de políticas permiten definir una política una vez y, después, utilizarla con varios principios y recursos. Las actualizaciones de la plantilla de política se reflejan en todas las políticas que utilizan la plantilla. Para obtener información más detallada consulte Plantillas de política de Cedar en la Guía de referencia sobre el lenguaje de las políticas de Cedar.

Por ejemplo, la siguiente plantilla de política proporciona `Read` `Comment` permisos y permisos para el director y el recurso que utilizan la plantilla de política. `Edit`

```
permit(
  principal == ?principal,
  action in [Action::"Read", Action::"Edit", Action::"Comment"],
  resource == ?resource
);
```

Si tuviera que crear una política con un nombre `Editor` basado en esta plantilla, cuando se designe a un director como editor de un recurso específico, su aplicación crearía una política que otorgue permisos al director para leer, editar y comentar el recurso.

A diferencia de las políticas estáticas, las políticas vinculadas a plantillas son dinámicas. Tomemos el ejemplo anterior: si eliminara la `Comment` acción de la plantilla de política, cualquier política vinculada a esa plantilla o basada en ella se actualizaría en consecuencia y los directores especificados en las políticas ya no podrían hacer comentarios sobre los recursos correspondientes.

Para ver más ejemplos de políticas vinculadas a plantillas, consulte. Ejemplos de políticas vinculadas a plantillas de permisos verificados de Amazon

Creación de plantillas de políticas de permisos verificados de Amazon

Puede crear plantillas de políticas en Verified Permissions utilizando el AWS Management Console AWS CLI, el o el AWSSDKs. Las plantillas de políticas permiten definir una política una vez y, a continuación, utilizarla con varios principios y recursos. Una vez que haya creado una plantilla de política, podrá crear políticas vinculadas a plantillas para utilizarlas con principios y recursos específicos. Para obtener más información, consulte Creación de políticas vinculadas a plantillas de permisos verificados de Amazon.

AWS Management Console

Para crear una plantilla de política

1. Abra la consola de permisos verificados en. https://console.aws.amazon.com/verifiedpermissions/ Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Plantillas de política.
3. Elija Crear plantilla de política.
4. En la sección Detalles, escriba una descripción de la plantilla de política.
5. En la sección Cuerpo de la plantilla de política, utilice los marcadores de posición `?principal` y `?resource` para permitir que las políticas creadas a partir de esta plantilla personalicen los permisos que conceden. Puede elegir Formato para dar formato a la sintaxis de su plantilla de política con el espaciado y la sangría recomendados.
6. Elija Crear plantilla de política.

AWS CLI

Para crear una plantilla de política

Puede crear una plantilla de políticas mediante la CreatePolicyTemplateoperación. En el siguiente ejemplo, se crea una plantilla de política con un marcador de posición para la entidad principal.

El archivo `template1.txt` contiene lo siguiente.

```
"VacationAccess"
permit(
    principal in ?principal,
```

```
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

```
$ aws verifiedpermissions create-policy-template \
    --description "Template for vacation picture access"
    --statement file://template1.txt
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-18T21:17:47.284268+00:00",
    "lastUpdatedDate": "2023-05-18T21:17:47.284268+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

Creación de políticas vinculadas a plantillas de permisos verificados de Amazon

Puede crear políticas vinculadas a plantillas, o políticas que se basen en una plantilla de política, utilizando, o. AWS Management Console AWS CLI AWS SDKs Las políticas vinculadas a plantillas permanecen vinculadas a sus plantillas de políticas. Si cambia la declaración de política en la plantilla de política, cualquier política vinculada a esa plantilla utilizará automáticamente la nueva declaración para todas las decisiones de autorización que se tomen a partir de ese momento.

Para ver ejemplos de políticas vinculadas a plantillas, consulte. Ejemplos de políticas vinculadas a plantillas de permisos verificados de Amazon

AWS Management Console

Para crear una política vinculada a una plantilla mediante la creación de una instancia de plantilla de política

1. Abra la consola de permisos verificados en. https://console.aws.amazon.com/verifiedpermissions/ Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Políticas.
3. Seleccione Crear política y, a continuación, elija Crear política vinculada a una plantilla.
4. Seleccione el botón de opción situado junto a la plantilla de política que desee utilizar y, a continuación, elija Siguiente.

5. Escriba la entidad principal y el recurso que desea utilizar para esta instancia específica de la política vinculada a la plantilla. Los valores especificados se muestran en el campo de vista previa de la instrucción de política.

   > **Note**
   >
   > Los valores de Entidad principal y Recurso deben tener el mismo formato que las políticas estáticas. Por ejemplo, para especificar el grupo `AdminUsers` de la entidad principal, escriba `Group::"AdminUsers"`. Si escribe `AdminUsers`, se muestra un error de validación.

6. Seleccione Crear política vinculada a una plantilla.

   La nueva política vinculada a una plantilla se muestra en Políticas.

AWS CLI

Para crear una política vinculada a una plantilla mediante la creación de una instancia de plantilla de política

Puede crear una política vinculada a una plantilla que haga referencia a una plantilla de política existente y que especifique los valores de cualquier marcador de posición utilizado por la plantilla.

En el siguiente ejemplo, se crea una política vinculada a una plantilla que utiliza una plantilla con la siguiente instrucción:

```
permit(
    principal in ?principal,
    action == PhotoFlash::Action::"view",
    resource == PhotoFlash::Photo::"VacationPhoto94.jpg"
);
```

También utiliza el siguiente archivo `definition.txt` para proporcionar el valor del parámetro `definition`:

```
{
    "templateLinked": {
        "policyTemplateId": "PTEXAMPLEabcdefg111111",
        "principal": {
```

```
            "entityType": "PhotoFlash::User",
            "entityId": "alice"
        }
    }
}
```

La salida muestra tanto el recurso, que se obtiene de la plantilla, como la entidad principal, que se obtiene del parámetro de definición.

```
$ aws verifiedpermissions create-policy \
    --definition file://definition.txt
    --policy-store-id PSEXAMPLEabcdefg111111
{
    "createdDate": "2023-05-22T18:57:53.298278+00:00",
    "lastUpdatedDate": "2023-05-22T18:57:53.298278+00:00",
    "policyId": "TPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATELINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "PhotoFlash::User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "PhotoFlash::Photo"
    }
}
```

Edición de plantillas de política de permisos verificados de Amazon

Puede editar o actualizar las plantillas de políticas en Verified Permissions utilizando el AWS Management Console AWS CLI, el o el AWS SDKs. Al editar una plantilla de políticas, se actualizarán automáticamente las políticas vinculadas a la plantilla o basadas en ella, así que tenga cuidado al editar las plantillas de políticas y asegúrese de no introducir accidentalmente un cambio que interrumpa su solicitud.

AWS Management Console

Para editar sus plantillas de política

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Plantillas de política. La consola muestra todas las plantillas de política que creó en el almacén de políticas actual.
3. Pulse el botón de opción situado junto a una plantilla de política para ver los detalles de la plantilla de política, como cuándo se creó, se actualizó y su contenido.
4. Seleccione Editar para editar sus plantillas de política. Actualice la descripción y el cuerpo de la política según sea necesario y, a continuación, seleccione Actualizar la plantilla de política.
5. Para eliminar una plantilla de política, pulse el botón de opción situado junto a la plantilla de política y, a continuación, seleccione Eliminar. Pulse Aceptar para confirmar la eliminación de la plantilla de política.

AWS CLI

Para actualizar una plantilla de política

Puede crear una política estática mediante la UpdatePolicyoperación. En el siguiente ejemplo, la plantilla de política especificada se actualiza sustituyendo su cuerpo de política por una nueva política definida en un archivo.

Contenido del archivo `template1.txt`:

```
permit(
    principal in ?principal,
    action == Action::"view",
    resource in ?resource)
when {
    principal has department && principal.department == "research"
};
```

```
$ aws verifiedpermissions update-policy-template \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --description "My updated template description" \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

```
{
    "createdDate": "2023-05-17T18:58:48.795411+00:00",
    "lastUpdatedDate": "2023-05-17T19:18:48.870209+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

Ejemplos de políticas vinculadas a plantillas de permisos verificados de Amazon

Al crear un almacén de políticas en Verified Permissions mediante el método de almacén de políticas de muestra, el almacén de políticas se crea con políticas predefinidas, plantillas de políticas y un esquema para el proyecto de ejemplo que haya elegido. Los siguientes ejemplos de políticas vinculadas a plantillas de Verified Permissions se pueden utilizar con los almacenes de políticas de muestra y sus políticas, plantillas de políticas y esquemas respectivos.

PhotoFlashejemplos

El siguiente ejemplo muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política Conceder acceso limitado a las fotos no privadas compartidas con un usuario y una foto individuales.

> **Note**
>
> El lenguaje de las políticas de Cedar considera que una entidad está `in`. Por lo tanto, `principal in User::"Alice"` es equivalente a `principal == User::"Alice"`.

```
permit (
 principal in PhotoFlash::User::"Alice",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Photo::"VacationPhoto94.jpg"
 );
```

En el siguiente ejemplo, se muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política Conceder acceso limitado a las fotos compartidas de carácter no privado con un usuario y un álbum individuales.

```
permit (
 principal in PhotoFlash::User::"Alice",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Album::"Italy2023"
 );
```

El siguiente ejemplo muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política Conceder acceso limitado a las fotos no privadas compartidas con un grupo de amigos y una foto individual.

```
permit (
 principal in PhotoFlash::FriendGroup::"Jane::MySchoolFriends",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Photo::"VacationPhoto94.jpg"
 );
```

En el siguiente ejemplo, se muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política Conceder acceso limitado a las fotos compartidas de carácter no privado con un grupo de amigos y un álbum.

```
permit (
 principal in PhotoFlash::FriendGroup::"Jane::MySchoolFriends",
 action in PhotoFlash::Action::"SharePhotoLimitedAccess",
 resource in PhotoFlash::Album::"Italy2023"
 );
```

En el siguiente ejemplo, se muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política Conceder acceso total a las fotos compartidas no privadas con un grupo de amigos y una foto individual.

```
permit (
 principal in PhotoFlash::UserGroup::"Jane::MySchoolFriends",
 action in PhotoFlash::Action::"SharePhotoFullAccess",
 resource in PhotoFlash::Photo::"VacationPhoto94.jpg"
 );
```

En el siguiente ejemplo, se muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política Bloquear a un usuario de una cuenta.

```
forbid(
```

```
 principal == PhotoFlash::User::"Bob",
 action,
 resource in PhotoFlash::Account::"Alice-account"
 );
```

DigitalPetStore ejemplos

El almacén de políticas de DigitalPetStore muestra no incluye ninguna plantilla de políticas. Para ver las políticas incluidas en el almacén de políticas, seleccione Políticas en el panel de navegación de la izquierda después de crear el almacén de políticas de DigitalPetStoremuestra.

TinyToDoejemplos

El siguiente ejemplo muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política que da acceso al espectador a un usuario individual y a una lista de tareas.

```
permit (
    principal == TinyTodo::User::"https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_h2aKCU1ts|5ae0c4b1-6de8-4dff-b52e-158188686f31|bob",
    action in [TinyTodo::Action::"ReadList", TinyTodo::Action::"ListTasks"],
    resource == TinyTodo::List::"1"
);
```

El siguiente ejemplo muestra cómo se puede crear una política vinculada a una plantilla que utilice la plantilla de política que da acceso de editor a un usuario individual y a una lista de tareas.

```
permit (
    principal == TinyTodo::User::"https://cognito-idp.us-east-1.amazonaws.com/us-
east-1_h2aKCU1ts|5ae0c4b1-6de8-4dff-b52e-158188686f31|bob",
    action in [
        TinyTodo::Action::"ReadList",
        TinyTodo::Action::"UpdateList",
        TinyTodo::Action::"ListTasks",
        TinyTodo::Action::"CreateTask",
        TinyTodo::Action::"UpdateTask",
        TinyTodo::Action::"DeleteTask"
    ],
    resource == TinyTodo::List::"1"
);
```

Uso de Amazon Verified Permissions con proveedores de identidades

Una fuente de identidad es una representación de un proveedor de identidad externo (IdP) en Amazon Verified Permissions. Las fuentes de identidad proporcionan información de un usuario que se autenticó con un IdP que tiene una relación de confianza con su almacén de políticas. Cuando la aplicación realiza una solicitud de autorización con un token de una fuente de identidad, el almacén de políticas puede tomar decisiones de autorización a partir de las propiedades del usuario y los permisos de acceso. Las fuentes de identidad de Verified Permissions mejoran la autorización con una conexión directa al almacén de identidades central y al servicio de autenticación.

Puede usar proveedores de identidad () de OpenID Connect (OIDCIdPs) con permisos verificados. Su aplicación puede generar solicitudes de autorización con OIDC identidad (ID) o acceder a tokens JSON web (JWTs). Con los identificadores de identidad, Verified Permissions lee las declaraciones de los usuarios IDs y las considera fundamentales para el control de acceso basado en atributos (). ABAC Con los tokens de acceso, Verified Permissions interpreta al usuario IDs como principal y el resto de las afirmaciones como contexto. Con ambos tipos de token, puedes asignar una reclamación similar `groups` a un grupo principal y crear políticas que evalúen el control de acceso basado en roles (). RBAC

Puede añadir un grupo de usuarios de Amazon Cognito o un OIDC IdP de OpenID Connect () personalizado como fuente de identidad.

Temas

- Uso de fuentes de identidad de Amazon Cognito
- Trabajar con OIDC fuentes de identidad
- Validación de clientes y audiencias
- Autorización del lado del cliente para JWTs
- Crear fuentes de identidad de Amazon Verified Permissions
- Editar fuentes de identidad de Amazon Verified Permissions
- Asignación de tokens de proveedores de identidad al esquema

Uso de fuentes de identidad de Amazon Cognito

Verified Permissions trabaja en estrecha colaboración con los grupos de usuarios de Amazon Cognito. Amazon Cognito JWTs tiene una estructura predecible. Verified Permissions reconoce esta estructura y aprovecha al máximo la información que contiene. Por ejemplo, puede implementar un modelo de autorización de control de acceso (RBAC) basado en roles con fichas de identificación o de acceso.

Una nueva fuente de identidad de grupos de usuarios de Amazon Cognito requiere la siguiente información:

- El Región de AWS.
- El ID del grupo de usuarios.
- El tipo de entidad de usuario que desea asociar a su fuente de identidad, por ejemplo`MyCorp::User`.
- El tipo de entidad de grupo que desea asociar a su fuente de identidad, por ejemplo`MyCorp::UserGroup`.
- (Opcional) El cliente IDs de su grupo de usuarios al que desea autorizar para realizar solicitudes a su almacén de políticas.

Como los permisos verificados solo funcionan con grupos de usuarios de Amazon Cognito en la misma cuenta Cuenta de AWS, no puede especificar una fuente de identidad en otra cuenta. Verified Permissions establece el prefijo de la entidad (el identificador de la fuente de identidad al que debe hacer referencia en las políticas que actúan sobre los principios del grupo de usuarios) como el ID de su grupo de usuarios, por ejemplo. `us-west-2_EXAMPLE`

Las notificaciones de los tokens del grupo de usuarios pueden contener atributos, ámbitos, grupos, datos de clientes y personalizados. IDs Amazon Cognito JWTs tiene la capacidad de incluir una variedad de información que puede contribuir a las decisiones de autorización en los permisos verificados. Entre ellos se incluyen:

1. Reclamaciones de nombre de usuario y grupo con un prefijo `cognito:`
2. Atributos de usuario personalizados con un `custom: prefix`
3. Las reclamaciones personalizadas se añaden en tiempo de ejecución
4. OIDCreclamaciones estándar como `sub` y `email`

Tratamos estas reclamaciones en detalle y cómo gestionarlas en las políticas de permisos verificados, enAsignación de tokens de proveedores de identidad al esquema.

> **Important**
>
> Si bien puede revocar los tokens de Amazon Cognito antes de que caduquenJWTs, se consideran recursos apátridas que son autónomos con firma y validez. Se espera que los servicios que cumplen con el JSON Web Token RFC 7519 validen los tokens de forma remota y no están obligados a validarlos con el emisor. Esto significa que Verified Permissions puede conceder acceso en función de un token que se haya revocado o emitido para un usuario que luego se haya eliminado. Para reducir este riesgo, le recomendamos que cree tokens con una validez lo más corta posible y que revoque los tokens de actualización cuando desee eliminar la autorización para continuar con la sesión de un usuario.

Las políticas de Cedar para las fuentes de identidad de los grupos de usuarios de Verified Permissions utilizan una sintaxis especial para los nombres de las notificaciones que contienen caracteres distintos de los alfanuméricos y los guiones bajos (). _ Esto incluye las notificaciones de prefijos de grupos de usuarios que contienen un `:` carácter, como y. `cognito:username` `custom:department` Para escribir una condición de la póliza que haga referencia a la `custom:department` afirmación `cognito:username` o a la reclamación, escríbalas como `principal["cognito:username"]` y`principal["custom:department"]`, respectivamente.

> **Note**
>
> Si un token contiene una reclamación con un `custom:` prefijo `cognito:` o y un nombre de reclamación con el valor literal `cognito` o`custom`, una solicitud de autorización con un prefijo no IsAuthorizedWithTokense aceptará con un`ValidationException`.

En el siguiente ejemplo, se muestra cómo puede crear una política que haga referencia a algunas de las reclamaciones del grupo de usuarios de Amazon Cognito asociadas a un principal.

```
permit(
     principal == ExampleCo::User::"us-east-1_example|4fe90f4a-ref8d9-4033-
a750-4c8622d62fb6",
     action,
     resource == ExampleCo::Photo::"VacationPhoto94.jpg"
```

```
)
when {
     principal["cognito:username"]) == "alice" &&
     principal["custom:department"]) == "Finance"
};
```

Para obtener más información sobre la representación cartográfica de las reclamaciones, consulteAsignación de tokens de ID al esquema. Para obtener más información sobre la autorización de los usuarios de Amazon Cognito, consulte Autorización con permisos verificados de Amazon en la Guía para desarrolladores de Amazon Cognito.

Trabajar con OIDC fuentes de identidad

También puede configurar cualquier OIDC IdP de OpenID Connect () compatible como fuente de identidad de un almacén de políticas. OIDClos proveedores son similares a los grupos de usuarios de Amazon Cognito: se producen JWTs como producto de la autenticación. Para añadir un OIDC proveedor, debe proporcionar un emisor URL

Una nueva fuente de OIDC identidad requiere la siguiente información:

- El emisor. URL Los permisos verificados deben poder detectar un `.well-known/openid-configuration` punto final en este URL punto.
- El tipo de token que quieres usar en las solicitudes de autorización. En este caso, ha elegido el token de identidad.
- Por ejemplo, el tipo de entidad de usuario que desea asociar a su fuente de identidad`MyCorp::User`.
- El tipo de entidad de grupo que desea asociar a su fuente de identidad, por ejemplo`MyCorp::UserGroup`.
- Un ejemplo de token de ID o una definición de las afirmaciones del token de ID.
- El prefijo que desea aplicar a la entidad IDs de usuario y grupo. En CLI yAPI, puede elegir este prefijo. En los almacenes de políticas que se crean con la opción Configurar con API Gateway y una fuente de identidad o con la opción de configuración guiada, Verified Permissions asigna un prefijo con el nombre del emisor menos`https://`, por ejemplo. `MyCorp::User::"auth.example.com|a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"`

La autorización con fuentes de OIDC identidad utiliza las mismas API operaciones que las fuentes de identidad del grupo de usuarios: y. IsAuthorizedWithTokenBatchIsAuthorizedWithToken

En el siguiente ejemplo, se muestra cómo se puede crear una política que permita el acceso a los informes de fin de año a los empleados del departamento de contabilidad que tengan una clasificación confidencial y no estén en una oficina satélite. Verified Permissions obtiene estos atributos de las afirmaciones que figuran en el token de identificación del director.

```
permit(
     principal in MyCorp::UserGroup::"MyOIDCProvider|Accounting",
     action,
     resource in MyCorp::Folder::"YearEnd2024"
) when {
     principal.jobClassification == "Confidential" &&
     !(principal.location like "SatelliteOffice*")
};
```

Validación de clientes y audiencias

Al añadir una fuente de identidad a un almacén de políticas, Verified Permissions tiene opciones de configuración que comprueban que los identificadores de identidad y de acceso se utilizan según lo previsto. Esta validación se produce en el procesamiento de las `BatchIsAuthorizedWithToken` API solicitudes `IsAuthorizedWithToken` y en ellas. El comportamiento difiere entre los identificadores y los tokens de acceso, y entre Amazon Cognito y las fuentes de OIDC identidad. Con los proveedores de grupos de usuarios de Amazon Cognito, Verified Permissions puede validar el ID de cliente tanto en el identificador como en el token de acceso. Con OIDC los proveedores, Verified Permissions puede validar el ID del cliente en los tokens de ID y la audiencia en los tokens de acceso.

Un ID de cliente es un identificador asociado a una OIDC aplicación OAuth o aplicación que está configurada con el proveedor, por ejemplo`1example23456789`. Una audiencia es una URL ruta asociada a la parte de confianza prevista, o al destino, de la aplicación de destino, por ejemplo`https://myapplication.example.com`. La `aud` afirmación no siempre está asociada a la audiencia.

Verified Permissions valida la fuente de identidad, la audiencia y el cliente de la siguiente manera:

Amazon Cognito

- Los tokens de Amazon Cognito ID tienen una `aud` declaración que contiene el ID de cliente de la aplicación. Los tokens de acceso tienen una `client_id` declaración que también contiene el ID de cliente de la aplicación.

Cuando ingresas uno o más valores para la validación de la aplicación cliente en tu fuente de identidad, Verified Permissions compara esta lista de clientes IDs de aplicaciones con la afirmación del token de ID o la `aud` afirmación del token `client_id` de acceso. Los permisos verificados no validan la audiencia de una parte de confianza para las fuentes de identidad de Amazon URL Cognito.

OIDC

OIDCLos tokens de identificación tienen una `aud` declaración que contiene una lista de clientes. IDs Los tokens de acceso tienen una `aud` declaración que contiene URL la audiencia del token. Los tokens de acceso también tienen una `client_id` declaración que contiene el ID de cliente deseado.

Puedes introducir uno o más valores para la validación de la audiencia con un OIDC proveedor. Cuando eliges un tipo de token de identificación, Verified Permissions valida el ID del cliente y comprueba que al menos un miembro del cliente IDs de la `aud` reclamación coincide con un valor de validación de audiencia.

Verified Permissions valida la audiencia para los tokens de acceso y comprueba que la `aud` afirmación coincide con un valor de validación de audiencia. Este valor del token de acceso proviene principalmente de la `aud` reclamación, pero puede proceder de la `client_id` reclamación `cid` or si no existe ninguna `aud` reclamación. Consulta con tu IdP el formato y la afirmación de audiencia correctos.

Un ejemplo de valor de validación de audiencia de un token de identificación es`1example23456789`.

Un ejemplo de valor de validación de la audiencia del token de acceso es`https://myapplication.example.com`.

Autorización del lado del cliente para JWTs

Es posible que desee procesar los tokens JSON web de su aplicación y transferir sus solicitudes a Verified Permissions sin utilizar una fuente de identidad de un almacén de políticas. Puedes extraer los atributos de tu entidad de un token JSON web (JWT) y analizarlos para convertirlos en permisos verificados.

En este ejemplo se muestra cómo se puede llamar a los permisos verificados desde un OIDC ID.[1]

```
async function authorizeUsingJwtToken(jwtToken) {
```

```
    const payload = await verifier.verify(jwtToken);

    var principalEntity = {
        entityType: "PhotoFlash::User", // the application needs to fill in the
 relevant user type
        entityId: payload["sub"], // the application need to use the claim that
 represents the user-id
    };
    var resourceEntity = {
        entityType: "PhotoFlash::Photo", //the application needs to fill in the
 relevant resource type
        entityId: "jane_photo_123.jpg", // the application needs to fill in the
 relevant resource id
    };
    var action = {
        actionType: "PhotoFlash::Action", //the application needs to fill in the
 relevant action id
        actionId: "GetPhoto", //the application needs to fill in the relevant action
 type
    };
    var entities = {
        entityList: [],
    };
    entities.entityList.push(...getUserEntitiesFromToken(payload));
    var policyStoreId = "PSEXAMPLEabcdefg111111"; // set your own policy store id

    const authResult = await client
        .isAuthorized({
        policyStoreId: policyStoreId,
        principal: principalEntity,
        resource: resourceEntity,
        action: action,
        entities,
        })
        .promise();

    return authResult;

}

function getUserEntitiesFromToken(payload) {
  let attributes = {};
  let claimsNotPassedInEntities = ['aud', 'sub', 'exp', 'jti', 'iss'];
```

```
  Object.entries(payload).forEach(([key, value]) => {
    if (claimsNotPassedInEntities.includes(key)) {
        return;
    }
    if (Array.isArray(value)) {
      var attibuteItem = [];
      value.forEach((item) => {
        attibuteItem.push({
          string: item,
        });
      });
      attributes[key] = {
        set: attibuteItem,
      };
    } else if (typeof value === 'string') {
      attributes[key] = {
        string: value,
      }
    } else if (typeof value === 'bigint' || typeof value ==='number') {
        attributes[key] = {
            long: value,
          }
    } else if (typeof value === 'boolean') {
        attributes[key] = {
            boolean: value,
       }
    }

  });

  let entityItem = {
    attributes: attributes,
    identifier: {
      entityType: "PhotoFlash::User",
      entityId: payload["sub"], // the application needs to use the claim that
 represents the user-id
    }
  };
  return [entityItem];
}
```

[1] Este ejemplo de código usa la aws-jwt-verifybiblioteca para verificar JWTs firmados por -compatible. OIDC IdPs

Crear fuentes de identidad de Amazon Verified Permissions

El siguiente procedimiento agrega una fuente de identidad a un almacén de políticas existente. Tras añadir la fuente de identidad, debe añadir los atributos al esquema.

También puede crear una fuente de identidad al crear un nuevo almacén de políticas en la consola de permisos verificados. En este proceso, puede importar automáticamente las notificaciones de los tokens de su fuente de identidad a los atributos de la entidad. Elige la opción Configuración guiada o Configuración con API Gateway y un proveedor de identidad. Estas opciones también crean políticas iniciales.

> **Note**
>
> Las fuentes de identidad no están disponibles en el panel de navegación de la izquierda hasta que haya creado un almacén de políticas. Las fuentes de identidad que cree están asociadas al almacén de políticas actual.

Puede omitir el tipo de entidad principal al crear una fuente de identidad con create-identity-sourcelos permisos CreateIdentitySourceverificados AWS CLI o entre ellosAPI. Sin embargo, un tipo de entidad en blanco crea una fuente de identidad con un tipo de entidad de`AWS::Cognito`. El nombre de esta entidad no es compatible con el esquema del almacén de políticas. Para integrar las identidades de Amazon Cognito en su esquema de almacén de políticas, debe establecer el tipo de entidad principal en una entidad de almacén de políticas compatible.

Temas

- Fuente de identidad de Amazon Cognito
- OIDCfuente de identidad

Fuente de identidad de Amazon Cognito

AWS Management Console

Para crear una fuente de identidad de un grupo de usuarios de Amazon Cognito

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Fuentes de identidad.

3. Seleccione Crear fuente de identidad.
4. En Detalles del grupo de usuarios de Cognito, seleccione Región de AWS e introduzca el ID del grupo de usuarios para su fuente de identidad.
5. En Configuración principal, elija un tipo principal para la fuente de identidad. Las identidades de los grupos de usuarios de Amazon Cognito conectados se asignarán al tipo de entidad principal seleccionado.
6. En Configuración de grupo, seleccione Usar el grupo de Cognito si quiere mapear la notificación del grupo `cognito:groups` de usuarios. Elija un tipo de entidad que sea principal del tipo principal.
7. En Validación de la aplicación del cliente, elija si desea validar la aplicación del clienteIDs.

   - Para validar la aplicación clienteIDs, elija Aceptar solo los tokens con una aplicación cliente coincidente IDs. Elija Agregar nuevo ID de aplicación cliente para cada ID de aplicación cliente que desee validar. Para eliminar un ID de aplicación cliente que se haya agregado, elija Eliminar junto al ID de la aplicación cliente.
   - Seleccione No validar la aplicación cliente IDs si no desea validar la aplicación clienteIDs.
8. Seleccione Crear fuente de identidad.
9. Para poder hacer referencia a los atributos que extraiga de los tokens de acceso o de identidad en sus políticas de Cedar, debe actualizar su esquema para que Cedar sepa qué tipo de entidad principal crea su fuente de identidad. Esta incorporación al esquema debe incluir los atributos a los que desee hacer referencia en sus políticas de Cedar. Para obtener más información sobre cómo asignar los atributos del token de Amazon Cognito a los atributos de entidad principal de Cedar, consulte Asignación de tokens de proveedores de identidad al esquema.

   Al crear un almacén API de políticas vinculado, Verified Permissions consulta los atributos de usuario del grupo de usuarios y crea un esquema en el que el tipo principal se rellena con los atributos del grupo de usuarios.

AWS CLI

Para crear una fuente de identidad de un grupo de usuarios de Amazon Cognito

Puede crear una fuente de identidad mediante la CreateIdentitySourceoperación. El siguiente ejemplo crea una fuente de identidad que puede acceder a las identidades autenticadas de un grupo de usuarios de Amazon Cognito.

El siguiente archivo `config.txt` contiene los detalles del grupo de usuarios de Amazon Cognito para que los utilice el parámetro --configuration del comando `create-identity-source`.

```
{
    "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"],
        "groupConfiguration": {
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

Comando:

```
$ aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
    --principal-entity-type "User" \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Para poder hacer referencia a los atributos que extraiga de los tokens de acceso o de identidad en sus políticas de Cedar, debe actualizar su esquema para que Cedar sepa qué tipo de entidad principal crea su fuente de identidad. Esta incorporación al esquema debe incluir los atributos a los que desee hacer referencia en sus políticas de Cedar. Para obtener más información sobre cómo asignar los atributos del token de Amazon Cognito a los atributos de entidad principal de Cedar, consulte Asignación de tokens de proveedores de identidad al esquema.

Al crear un almacén API de políticas vinculado, Verified Permissions consulta los atributos de usuario del grupo de usuarios y crea un esquema en el que el tipo principal se rellena con los atributos del grupo de usuarios.

Para obtener más información sobre el uso de los tokens de acceso e identidad de Amazon Cognito para los usuarios autenticados en Verified Permissions, consulte Autorización con Amazon Verified Permissions en la Guía para desarrolladores de Amazon Cognito.

OIDCfuente de identidad

AWS Management Console

Para crear una fuente de identidad de OpenID Connect (OIDC)

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Fuentes de identidad.
3. Seleccione Crear fuente de identidad.
4. Elige un OIDCproveedor externo.
5. En Emisor URL, introduzca el URL de su OIDC emisor. Este es el punto final del servicio que proporciona el servidor de autorización, las claves de firma y otra información sobre su proveedor, por ejemplo. `https://auth.example.com` El emisor URL debe alojar un documento de OIDC descubrimiento en`/.well-known/openid-configuration`.
6. En Tipo de token, elija el tipo OIDC JWT que desea que envíe su solicitud de autorización. Para obtener más información, consulte Asignación de tokens de proveedores de identidad al esquema.
7. En Reclamaciones de usuario y grupo, elija una entidad de usuario y una reclamación de usuario como fuente de identidad. La entidad de usuario es una entidad de su almacén de políticas a la que quiere hacer referencia a los usuarios de su OIDC proveedor. La afirmación de usuario proviene, por lo general`sub`, de tu ID o token de acceso que contiene el identificador único de la entidad que se va a evaluar. Las identidades del OIDC IdP conectado se asignarán al tipo principal seleccionado.
8. En las notificaciones de usuario y grupo, elija una entidad de grupo y una reclamación de grupo como fuente de identidad. La entidad del grupo es la matriz de la entidad del usuario. Las reclamaciones grupales se asignan a esta entidad. La reclamación de grupo proviene, por lo general`groups`, de su ID o token de acceso que contiene una cadena o cadena delimitada por espacios de nombres de grupos de usuarios para la entidad que se va a evaluar. JSON Las identidades del OIDC IdP conectado se asignarán al tipo principal seleccionado.

9. En la validación de audiencia, introduzca el cliente IDs o la audiencia URLs que desea que su almacén de políticas acepte en las solicitudes de autorización, si las hubiera.
10. Seleccione Crear fuente de identidad.
11. Actualice su esquema para que Cedar conozca el tipo de principal que crea su fuente de identidad. Esta incorporación al esquema debe incluir los atributos a los que desee hacer referencia en sus políticas de Cedar. Para obtener más información sobre cómo asignar los atributos del token de Amazon Cognito a los atributos de entidad principal de Cedar, consulte Asignación de tokens de proveedores de identidad al esquema.

    Al crear un almacén API de políticas vinculado, Verified Permissions consulta los atributos de usuario del grupo de usuarios y crea un esquema en el que el tipo principal se rellena con los atributos del grupo de usuarios.

AWS CLI

Para crear una fuente de OIDC identidad

Puede crear una fuente de identidad mediante la CreateIdentitySourceoperación. El siguiente ejemplo crea una fuente de identidad que puede acceder a las identidades autenticadas de un grupo de usuarios de Amazon Cognito.

El siguiente `config.txt` archivo contiene los detalles de un OIDC IdP para que los utilice el `--configuration` parámetro del `create-identity-source` comando. En este ejemplo, se crea una fuente de OIDC identidad para los tokens de identificación.

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth.example.com",
        "tokenSelection": {
                "identityTokenOnly": {
                        "clientIds":["1example23456789"],
                        "principalIdClaim": "sub"
                },
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
```

```
}
```

El siguiente `config.txt` archivo contiene los detalles de un OIDC IdP para que los utilice el `--configuration` parámetro del `create-identity-source` comando. En este ejemplo, se crea una fuente de OIDC identidad para los tokens de acceso.

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth.example.com",
        "tokenSelection": {
                "accessTokenOnly": {
                        "audiences":["https://auth.example.com"],
                        "principalIdClaim": "sub"
                },
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

Comando:

```
$ aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
    --principal-entity-type "User" \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Para poder hacer referencia a los atributos que extraiga de los tokens de acceso o de identidad en sus políticas de Cedar, debe actualizar su esquema para que Cedar sepa qué tipo de entidad principal crea su fuente de identidad. Esta incorporación al esquema debe incluir los atributos a los que desee hacer referencia en sus políticas de Cedar. Para obtener más información sobre

cómo asignar los atributos del token de Amazon Cognito a los atributos de entidad principal de Cedar, consulte Asignación de tokens de proveedores de identidad al esquema.

Al crear un almacén API de políticas vinculado, Verified Permissions consulta los atributos de usuario del grupo de usuarios y crea un esquema en el que el tipo principal se rellena con los atributos del grupo de usuarios.

Editar fuentes de identidad de Amazon Verified Permissions

Puede editar algunos parámetros de su fuente de identidad después de crearla. Si el esquema del almacén de políticas coincide con los atributos de la fuente de identidad, tenga en cuenta que debe actualizar el esquema por separado para reflejar los cambios que realice en la fuente de identidad.

Temas

- Fuente de identidad de los grupos de usuarios de Amazon Cognito
- Fuente de identidad de OpenID Connect (OIDC)

Fuente de identidad de los grupos de usuarios de Amazon Cognito

AWS Management Console

Para actualizar una fuente de identidad de un grupo de usuarios de Amazon Cognito

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Fuentes de identidad.
3. Seleccione el ID de la fuente de identidad que desee editar.
4. Elija Editar.
5. En Detalles del grupo de usuarios de Cognito, seleccione Región de AWS y escriba el ID del grupo de usuarios para su fuente de identidad.
6. En Detalles principales, puede actualizar el tipo principal de la fuente de identidad. Las identidades de los grupos de usuarios de Amazon Cognito conectados se asignarán al tipo de entidad principal seleccionado.
7. En Configuración de grupo, seleccione Usar el grupo de Cognito si quiere mapear la notificación del grupo `cognito:groups` de usuarios. Elija un tipo de entidad que sea principal del tipo principal.

8. En Validación de la aplicación del cliente, elija si desea validar la aplicación del clienteIDs.
   - Para validar la aplicación clienteIDs, elija Aceptar solo los tokens con una aplicación cliente coincidente IDs. Elija Agregar nuevo ID de aplicación cliente para cada ID de aplicación cliente que desee validar. Para eliminar un ID de aplicación cliente que se haya agregado, elija Eliminar junto al ID de la aplicación cliente.
   - Seleccione No validar la aplicación cliente IDs si no desea validar la aplicación clienteIDs.
9. Elija Guardar cambios.
10. Si ha cambiado el tipo de entidad principal de la fuente de identidad, debe actualizar el esquema para que refleje correctamente el tipo de entidad principal actualizado.

Para eliminar una fuente de identidad, pulse el botón de opción situado junto a una fuente de identidad y, a continuación, elija Eliminar fuente de identidad. Escriba `delete` en el cuadro de texto y, a continuación, seleccione Eliminar fuente de identidad para confirmar la eliminación de la fuente de identidad.

AWS CLI

Para actualizar una fuente de identidad de un grupo de usuarios de Amazon Cognito

Puede actualizar una fuente de identidad mediante la UpdateIdentitySourceoperación. El siguiente ejemplo actualiza la fuente de identidad especificada para usar un grupo de usuarios de Amazon Cognito diferente.

El siguiente archivo `config.txt` contiene los detalles del grupo de usuarios de Amazon Cognito para que los utilice el parámetro --configuration del comando `create-identity-source`.

```
{
    "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"],
        "groupConfiguration": {
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

Comando:

```
$ aws verifiedpermissions update-identity-source \
    --update-configuration file://config.txt \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Si cambia el tipo de entidad principal de la fuente de identidad, debe actualizar el esquema para que refleje correctamente el tipo de entidad principal actualizado.

Fuente de identidad de OpenID Connect (OIDC)

AWS Management Console

Para actualizar una fuente de OIDC identidad

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Elige tu almacén de pólizas.
2. En el panel de navegación de la izquierda, elija Fuentes de identidad.
3. Seleccione el ID de la fuente de identidad que desee editar.
4. Elija Editar.
5. En los detalles del OIDC proveedor, cambia el emisor URL según sea necesario.
6. En las reclamaciones de los tokens de mapa con los atributos del esquema, cambie las asociaciones entre las afirmaciones de usuario y grupo y los tipos de entidad del almacén de políticas, según sea necesario. Después de cambiar los tipos de entidad, debe actualizar las políticas y los atributos del esquema para aplicarlos a los nuevos tipos de entidad.
7. En la validación de audiencia, añade o elimina los valores de audiencia que quieras aplicar.
8. Elija Guardar cambios.

Para eliminar una fuente de identidad, pulse el botón de opción situado junto a una fuente de identidad y, a continuación, elija Eliminar fuente de identidad. Escriba `delete` en el cuadro de texto y, a continuación, seleccione Eliminar fuente de identidad para confirmar la eliminación de la fuente de identidad.

AWS CLI

Para actualizar una fuente OIDC de identidad

Puede actualizar una fuente de identidad mediante la UpdateIdentitySourceoperación. En el siguiente ejemplo, se actualiza la fuente de identidad especificada para que utilice un OIDC proveedor diferente.

El siguiente archivo `config.txt` contiene los detalles del grupo de usuarios de Amazon Cognito para que los utilice el parámetro --configuration del comando `create-identity-source`.

```
{
    "openIdConnectConfiguration": {
        "issuer": "https://auth2.example.com",
        "tokenSelection": {
                "identityTokenOnly": {
                        "clientIds":["2example10111213"],
                        "principalIdClaim": "sub"
                },
        },
        "entityIdPrefix": "MyOIDCProvider",
        "groupConfiguration": {
              "groupClaim": "groups",
              "groupEntityType": "MyCorp::UserGroup"
        }
    }
}
```

Comando:

```
$ aws verifiedpermissions update-identity-source \
    --update-configuration file://config.txt \
    --policy-store-id 123456789012
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Si cambia el tipo de entidad principal de la fuente de identidad, debe actualizar el esquema para que refleje correctamente el tipo de entidad principal actualizado.

Asignación de tokens de proveedores de identidad al esquema

Es posible que desee añadir una fuente de identidad a un almacén de políticas y las reclamaciones del proveedor de mapas a su esquema de almacén de políticas. Puedes automatizar este proceso o actualizar el esquema manualmente. Una vez que haya asignado los tokens al esquema, puede crear políticas que hagan referencia a ellos.

Esta sección de la guía del usuario contiene la siguiente información:

- Cuándo puede rellenar automáticamente los atributos de un esquema de almacén de políticas
- Cómo usar Amazon Cognito y las reclamaciones de OIDC token en tus políticas de permisos verificados
- ¿Cómo crear manualmente un esquema para una fuente de identidad

APILos almacenes de políticas enlazados y los almacenes de políticas con una fuente de identidad mediante la configuración guiada no requieren la asignación manual de los atributos del token de identidad (ID) al esquema. Puedes proporcionar permisos verificados con los atributos de tu grupo de usuarios o de tus OIDC tokens y crear un esquema que incluya los atributos de los usuarios. En la autorización de los tokens de identificación, Verified Permissions asigna las reclamaciones a los atributos de una entidad principal. Es posible que tenga que asignar manualmente los tokens de Amazon Cognito a su esquema en las siguientes condiciones:

- Creó un almacén de políticas o un almacén de políticas en blanco a partir de una muestra.
- Desea extender el uso de los tokens de acceso más allá del control de acceso basado en roles ()RBAC.
- Los almacenes de políticas se crean con los permisos verificados REST API AWS SDK, un o los. AWS CDK

Para usar Amazon Cognito o un proveedor de OIDC identidad (IdP) como fuente de identidad en su almacén de políticas de permisos verificados, debe tener atributos de proveedor en su esquema. Si creó su almacén de políticas de forma que rellene automáticamente su esquema a partir de la información del proveedor en un token de identificación, está listo para escribir políticas. Si crea un almacén de políticas sin un esquema para su fuente de identidad, debe agregar los atributos del proveedor al esquema. El esquema debe corresponder a las entidades en las que los tokens del proveedor crean IsAuthorizedWithTokeno BatchIsAuthorizedWithTokenAPIsolicitan. A continuación, puede escribir políticas utilizando los atributos del token del proveedor.

Para obtener más información sobre el uso del ID de Amazon Cognito y los tokens de acceso para los usuarios autenticados en los permisos verificados, consulte Autorización con permisos verificados de Amazon en la Guía para desarrolladores de Amazon Cognito.

Temas

Lo que debe saber sobre el mapeo de esquemas

El mapeo de atributos difiere entre los tipos de token

En la autorización del token de acceso, Verified Permissions asigna las reclamaciones al contexto. En la autorización del token de identificación, Verified Permissions asigna las reclamaciones a los atributos principales. En el caso de los almacenes de políticas que cree en la consola de permisos verificados, solo los almacenes de políticas vacíos y de ejemplo no tienen una fuente de identidad y requieren que complete el esquema con los atributos del grupo de usuarios para la autorización del token de identificación. La autorización de los tokens de acceso se basa en un control de acceso basado en roles (RBAC) con solicitudes de pertenencia a grupos y no asigna automáticamente otras solicitudes al esquema del almacén de políticas.

Los atributos de la fuente de identidad no son obligatorios

Al crear una fuente de identidad en la consola de permisos verificados, no se marca ningún atributo como obligatorio. Esto evita que las reclamaciones incumplidas provoquen errores de validación en las solicitudes de autorización. Puede establecer los atributos como obligatorios según sea necesario, pero deben estar presentes en todas las solicitudes de autorización.

RBACno requiere atributos en el esquema

Los esquemas de las fuentes de identidad dependen de las asociaciones de entidades que realice al agregar la fuente de identidad. Una fuente de identidad asigna una reclamación a un tipo de entidad de usuario y otra a un tipo de entidad de grupo. Estas asignaciones de entidades son el núcleo de una configuración de fuente de identidad. Con esta información mínima, puede escribir políticas que realicen acciones de autorización para usuarios específicos y grupos específicos de los que los usuarios puedan ser miembros, en un modelo de control de acceso basado en roles (). RBAC La

adición de notificaciones de token al esquema amplía el ámbito de autorización de su almacén de políticas. Los atributos de usuario de los tokens de identificación contienen información sobre los usuarios que puede contribuir a la autorización del control de acceso (ABAC) basado en atributos. Los atributos de contexto de los tokens de acceso tienen información similar a los ámbitos OAuth 2.0 que pueden aportar información adicional sobre el control de acceso por parte del proveedor, pero requieren modificaciones adicionales en el esquema.

Las opciones Configuración con API Gateway y una fuente de identidad y Configuración guiada de la consola de permisos verificados asignan las notificaciones de los tokens de identificación al esquema. Este no es el caso de las solicitudes de token de acceso. Para añadir a tu esquema notificaciones de token de acceso que no sean grupales, debes editarlo en JSON modo y añadir atributos. commonTypes Para obtener más información, consulte Asignar tokens de acceso.

OIDCgroup claim admite varios formatos

Al añadir un OIDC proveedor, puede elegir el nombre de la reclamación del grupo en el identificador o en los tokens de acceso que desea asignar a la pertenencia al grupo de un usuario en su almacén de políticas. Los permisos verificados reconocen las solicitudes de los grupos en los siguientes formatos:

1. Cadena sin espacios: `"groups": "MyGroup"`
2. Lista delimitada por espacios:. `"groups": "MyGroup1 MyGroup2 MyGroup3"` Cada cadena es un grupo.
3. JSONlista (delimitada por comas): `"groups": ["MyGroup1", "MyGroup2", "MyGroup3"]`

> **Note**
>
> Los permisos verificados interpretan cada cadena de una reclamación de grupo separada por espacios como un grupo independiente. Para interpretar el nombre de un grupo con un carácter de espacio como un grupo único, sustituya o elimine el espacio de la afirmación. Por ejemplo, formatee un grupo denominado `My Group` como`MyGroup`.

Elija un tipo de token

La forma en que el almacén de políticas trabaja con la fuente de identidad depende de una decisión clave en la configuración de la fuente de identidad: si va a procesar los tokens de identificación o de acceso. Con un proveedor de identidades de Amazon Cognito, puede elegir el tipo de token al

crear un almacén de políticas API vinculado a él. Al crear un almacén API de políticas vinculado a un servidor, debe elegir si desea configurar la autorización para los tokens de identificación o de acceso. Esta información afecta a los atributos del esquema que Verified Permissions aplica a su almacén de políticas y a la sintaxis del autorizador Lambda de su puerta de enlace. API API Con un OIDC proveedor, debe elegir un tipo de token al agregar la fuente de identidad. Puedes elegir un identificador o un token de acceso y, si eliges, no se procesará en tu almacén de políticas el tipo de token que no hayas elegido. Especialmente si quieres beneficiarte de la asignación automática de las solicitudes de token de identificación a los atributos de la consola de permisos verificados, decide con antelación qué tipo de token quieres procesar antes de crear tu fuente de identidad. Cambiar el tipo de token requiere un esfuerzo considerable para refactorizar las políticas y el esquema. En los siguientes temas se describe el uso de los identificadores de acceso y de identificación en los almacenes de pólizas.

El analizador Cedar requiere corchetes para algunos caracteres

Las políticas suelen hacer referencia a los atributos del esquema en un formato como`principal.username`. En el caso de la mayoría de los caracteres no alfanuméricos`:`, como`.`, o `/` que puedan aparecer en los nombres de las notificaciones de los tokens, Verified Permissions no puede analizar un valor de condición como `principal.cognito:username` o. `context.ip-address` En su lugar, debe formatear estas condiciones con una notación entre corchetes en el formato `principal["cognito:username"]` o`context["ip-address"]`, respectivamente. El carácter de subrayado `_` es un carácter válido en los nombres de las reclamaciones y es la única excepción no alfanumérica a este requisito.

Un ejemplo parcial de un esquema de un atributo principal de este tipo es el siguiente:

```
"User": {
   "shape": {
      "type": "Record",
      "attributes": {
         "cognito:username": {
            "type": "String",
            "required": true
         },
         "custom:employmentStoreCode": {
            "type": "String",
            "required": true,
         },
         "email": {
            "type": "String",
```

```
            "required": false
         }
      }
   }
}
```

Un ejemplo parcial de un esquema de un atributo de contexto de este tipo tiene el siguiente aspecto:

```
"GetOrder": {
   "memberOf": [],
   "appliesTo": {
      "resourceTypes": [
         "Order"
      ],
      "context": {
         "type": "Record",
         "attributes": {
            "ip-address": {
               "required": false,
               "type": "String"
            }
   }
   },
      "principalTypes": [
         "User"
      ]
   }
}
```

Para ver un ejemplo de política que se validará con este esquema, consulteUtiliza la notación entre corchetes para hacer referencia a los atributos del token.

Asignación de tokens de ID al esquema

Verified Permissions procesa las reclamaciones de los tokens de identificación como atributos del usuario: sus nombres y cargos, su pertenencia a un grupo y su información de contacto. Los tokens de identificación son especialmente útiles en un modelo de autorización de control de acceso (ABAC) basado en atributos. Si quieres que los permisos verificados analicen el acceso a los recursos en función de quién realiza la solicitud, elige los tokens de identificación para tu fuente de identidad.

Tokens de Amazon Cognito ID

Los tokens de Amazon Cognito ID funcionan con la mayoría de las bibliotecas de partes confiablesOIDC. Amplían las funciones con afirmaciones adicionales. OIDC La aplicación puede autenticar al usuario con las operaciones de API autenticación de los grupos de usuarios de Amazon Cognito o con la interfaz de usuario alojada en el grupo de usuarios. Para obtener más información, consulte Uso de los puntos de enlace API y en la Guía para desarrolladores de Amazon Cognito.

Afirmaciones útiles en los tokens de Amazon Cognito ID

`cognito:username` y *`preferred_username`*

Variantes del nombre de usuario del usuario.

`sub`

El identificador de usuario único del usuario (UUID)

Reclamaciones con *`custom:`* prefijo

Un prefijo para los atributos personalizados del grupo de usuarios, como. `custom:employmentStoreCode`

Reclamaciones estándar

OIDCReclamaciones estándar como `email` y`phone_number`. Para obtener más información, consulte las afirmaciones estándar de OpenID Connect Core 1.0 que incorporan el conjunto de erratas 2.

`cognito:groups`

Pertenencias a grupos de un usuario. En un modelo de autorización basado en el control de acceso basado en roles (RBAC), esta afirmación presenta las funciones que puede evaluar en sus políticas.

Reclamaciones transitorias

Reclamaciones que no son propiedad del usuario, pero que se agregan en tiempo de ejecución mediante un disparador Lambda previo a la generación del token. Las afirmaciones transitorias se parecen a las afirmaciones estándar, pero están fuera de la norma, por ejemplo`tenant`, o. `department`

En las políticas que hacen referencia a los atributos de Amazon Cognito que tienen un `:` separador, haga referencia a los atributos en el formato. `principal["`*`cognito:username`*`"]` La afirmación

de los roles `cognito:groups` es una excepción a esta regla. Verified Permissions asigna el contenido de esta declaración a las entidades principales de la entidad de usuario.

Para obtener más información sobre la estructura de los tokens de ID de los grupos de usuarios de Amazon Cognito, consulte Uso del token de ID en la Guía para desarrolladores de Amazon Cognito.

El siguiente ejemplo de token de identificación tiene cada uno de los cuatro tipos de atributos. Incluye la notificación específica de Amazon Cognito `cognito:username`, la notificación personalizada `custom:employmentStoreCode`, la notificación estándar `email`, y la notificación transitoria `tenant`.

```
{
    "sub": "91eb4550-XXX",
    "cognito:groups": [
        "Store-Owner-Role",
        "Customer"
    ],
    "email_verified": true,
    "clearance": "confidential",
    "iss": "https://cognito-idp.us-east-2.amazonaws.com/us-east-2_EXAMPLE",
    "cognito:username": "alice",
    "custom:employmentStoreCode": "petstore-dallas",
    "origin_jti": "5b9f50a3-05da-454a-8b99-b79c2349de77",
    "aud": "1example23456789",
    "event_id": "0ed5ad5c-7182-4ecf-XXX",
    "token_use": "id",
    "auth_time": 1687885407,
    "department": "engineering",
    "exp": 1687889006,
    "iat": 1687885407,
    "tenant": "x11app-tenant-1",
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "email": "alice@example.com"
}
```

Cuando crea una fuente de identidad con su grupo de usuarios de Amazon Cognito, especifica el tipo de entidad principal con la que Verified Permissions genera las solicitudes de autorización. `IsAuthorizedWithToken` Después, sus políticas pueden probar los atributos de esa entidad principal como parte de la evaluación de esa solicitud. Su esquema define el tipo y los atributos principales de una fuente de identidad y, a continuación, puede hacer referencia a ellos en sus políticas de Cedar.

También especifica el tipo de entidad de grupo que desea obtener de la afirmación del grupo de fichas de identificación. En las solicitudes de autorización, Verified Permissions asigna cada miembro de la reclamación del grupo a ese tipo de entidad de grupo. En las políticas, puede hacer referencia a esa entidad del grupo como principal.

El siguiente ejemplo muestra cómo reflejar los atributos del token de identidad de ejemplo en su esquema de Verified Permissions. Para obtener más información sobre cómo editar el esquema, consulte Edición de esquemas de almacenes de políticas en modo JSON. Si la configuración de su fuente de identidad especifica el tipo de entidad principal `User`, puede incluir algo similar al siguiente ejemplo para que esos atributos estén disponibles para Cedar.

```
"User": {
   "shape": {
      "type": "Record",
      "attributes": {
         "cognito:username": {
            "type": "String",
            "required": false
         },
         "custom:employmentStoreCode": {
            "type": "String",
            "required": false
         },
         "email": {
            "type": "String"
         },
         "tenant": {
            "type": "String",
            "required": true
         }
      }
   }
}
```

Para ver un ejemplo de política que se validará con este esquema, consulteRefleja los atributos del token de Amazon Cognito ID.

OIDCTokens de identificación

Trabajar con los tokens de ID de un OIDC proveedor es muy parecido a trabajar con los tokens de ID de Amazon Cognito. La diferencia está en las afirmaciones. Su IdP puede presentar OIDCatributos

estándar o tener un esquema personalizado. Al crear un nuevo almacén de políticas en la consola de permisos verificados, puede agregar una fuente de OIDC identidad con un token de ID de ejemplo o puede asignar manualmente las notificaciones de los tokens a los atributos del usuario. Como Verified Permissions no conoce el esquema de atributos de tu IdP, debes proporcionar esta información.

Para obtener más información, consulte Crear almacenes de políticas de Verified Permissions.

El siguiente es un ejemplo de esquema para un almacén de políticas con una fuente de OIDC identidad.

```
"User": {
   "shape": {
      "type": "Record",
      "attributes": {
         "email": {
            "type": "String"
         },
         "email_verified": {
            "type": "Boolean"
         },
         "name": {
            "type": "String",
            "required": true
         },
         "phone_number": {
            "type": "String"
         },
         "phone_number_verified": {
            "type": "Boolean"
         }
      }
   }
}
```

Para ver un ejemplo de política que se validará con este esquema, consulteRefleja los atributos OIDC del token de ID.

Asignar tokens de acceso

Verified Permissions procesa las notificaciones de token de acceso distintas de las declaradas por el grupo como atributos de la acción o atributos de contexto. Además de la pertenencia a un grupo,

los tokens de acceso de su IdP pueden contener información sobre API el acceso. Los tokens de acceso son útiles en los modelos de autorización que utilizan el control de acceso basado en roles (). RBAC Los modelos de autorización que se basan en declaraciones de token de acceso distintas de la pertenencia a un grupo requieren un esfuerzo adicional en la configuración del esquema.

Asignar tokens de acceso de Amazon Cognito

Los tokens de acceso de Amazon Cognito tienen notificaciones que se pueden usar en las autorizaciones:

Afirmaciones útiles en los tokens de acceso de Amazon Cognito

`client_id`

El ID de la aplicación cliente de la parte que OIDC confía. Con el ID de cliente, Verified Permissions puede comprobar que la solicitud de autorización proviene de un cliente autorizado para el almacén de políticas. En la autorización machine-to-machine (M2M), el sistema solicitante autoriza una solicitud con un secreto de cliente y proporciona el identificador del cliente y los alcances como prueba de la autorización.

`scope`

Los ámbitos OAuth 2.0 que representan los permisos de acceso del portador del token.

`cognito:groups`

Pertenencias a grupos de un usuario. En un modelo de autorización basado en el control de acceso basado en roles (RBAC), esta afirmación presenta las funciones que puede evaluar en sus políticas.

Reclamaciones transitorias

Reclamaciones que no son un permiso de acceso, pero que se añaden en tiempo de ejecución mediante un disparador Lambda previo a la generación del token de un grupo de usuarios. Las afirmaciones transitorias se parecen a las afirmaciones estándar, pero están fuera del estándar, por ejemplo`tenant`, o. `department` La personalización de los tokens de acceso añade un coste a tu AWS factura.

Para obtener más información sobre la estructura de los tokens de acceso de los grupos de usuarios de Amazon Cognito, consulte Uso del token de acceso en la Guía para desarrolladores de Amazon Cognito.

Un token de acceso de Amazon Cognito se asigna a un objeto de contexto cuando se transfiere a Verified Permissions. Se puede hacer referencia a los atributos del token de acceso mediante `context.token.`*`attribute_name`*. El siguiente ejemplo de token de acceso incluye tanto el `client_id` como las notificaciones de `scope`.

```
{
    "sub": "91eb4550-9091-708c-a7a6-9758ef8b6b1e",
    "cognito:groups": [
        "Store-Owner-Role",
        "Customer"
    ],
    "iss": "https://cognito-idp.us-east-2.amazonaws.com/us-east-2_EXAMPLE",
    "client_id": "1example23456789",
    "origin_jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "event_id": "bda909cb-3e29-4bb8-83e3-ce6808f49011",
    "token_use": "access",
    "scope": "MyAPI/mydata.write",
    "auth_time": 1688092966,
    "exp": 1688096566,
    "iat": 1688092966,
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN2222222",
    "username": "alice"
}
```

El siguiente ejemplo muestra cómo reflejar los atributos del token de acceso de ejemplo en su esquema de Verified Permissions. Para obtener más información sobre cómo editar el esquema, consulte Edición de esquemas de almacenes de políticas en modo JSON.

```
{
   "MyApplication": {
      "actions": {
         "Read": {
            "appliesTo": {
               "context": {
                  "type": "ReusedContext"
               },
               "resourceTypes": [
                  "Application"
               ],
               "principalTypes": [
                  "User"
               ]
```

```
            }
         }
      },
      ...
      ...
      "commonTypes": {
         "ReusedContext": {
            "attributes": {
               "token": {
                  "type": "Record",
                  "attributes": {
                     "scope": {
                        "type": "Set",
                        "element": {
                           "type": "String"
                        }
                     },
                     "client_id": {
                        "type": "String"
                     }
                  }
               }
            },
            "type": "Record"
         }
      }
   }
}
```

Para ver un ejemplo de política que se validará con este esquema, consulte. Refleja los atributos del token de acceso de Amazon Cognito

Mapeo de OIDC los tokens de acceso

La mayoría de los tokens de acceso de OIDC proveedores externos se alinean estrechamente con los tokens de acceso de Amazon Cognito. Un token de OIDC acceso se asigna a un objeto de contexto cuando se pasa a Verified Permissions. Se puede hacer referencia a los atributos del token de acceso mediante `context.token.`*`attribute_name`*. El siguiente ejemplo de token de OIDC acceso incluye ejemplos de notificaciones base.

```
{
    "sub": "91eb4550-9091-708c-a7a6-9758ef8b6b1e",
```

```
    "groups": [
        "Store-Owner-Role",
        "Customer"
    ],
    "iss": "https://auth.example.com",
    "client_id": "1example23456789",
    "aud": "https://myapplication.example.com"
    "scope": "MyAPI-Read",
    "exp": 1688096566,
    "iat": 1688092966,
    "jti": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN2222222",
    "username": "alice"
}
```

El siguiente ejemplo muestra cómo reflejar los atributos del token de acceso de ejemplo en su esquema de Verified Permissions. Para obtener más información sobre cómo editar el esquema, consulte Edición de esquemas de almacenes de políticas en modo JSON.

```
{
   "MyApplication": {
      "actions": {
         "Read": {
            "appliesTo": {
               "context": {
                  "type": "ReusedContext"
               },
               "resourceTypes": [
                  "Application"
               ],
               "principalTypes": [
                  "User"
               ]
            }
         }
      },
      ...
      ...
      "commonTypes": {
         "ReusedContext": {
            "attributes": {
               "token": {
                  "type": "Record",
                  "attributes": {
```

```
                    "scope": {
                       "type": "Set",
                       "element": {
                          "type": "String"
                       }
                    },
                    "client_id": {
                       "type": "String"
                    }
                 }
              }
           },
           "type": "Record"
        }
     }
  }
}
```

Para ver un ejemplo de política que se validará con este esquema, consulteRefleja OIDC los atributos del token de acceso.

Notación alternativa para las reclamaciones delimitadas por dos puntos de Amazon Cognito

En el momento en que se lanzó Verified Permissions, el esquema recomendado para el token de Amazon Cognito afirmaba lo mismo `cognito:groups` y `custom:store` convertía estas cadenas delimitadas por dos puntos para utilizar el `.` carácter como delimitador jerárquico. Este formato se denomina notación de puntos. Por ejemplo, una referencia a lo que `cognito:groups` pasó a figurar `principal.cognito.groups` en sus políticas. Aunque puede seguir utilizando este formato, le recomendamos que cree su esquema y sus políticas con una notación entre corchetes. En este formato, una referencia `cognito:groups` se convierte `principal["cognito:groups"]` en una referencia en sus políticas. Los esquemas generados automáticamente para los identificadores de los grupos de usuarios desde la consola de permisos verificados utilizan la notación entre corchetes.

Puede seguir utilizando la notación de puntos en los esquemas y políticas creados manualmente para las fuentes de identidad de Amazon Cognito. No puede usar la notación de puntos `:` ni ningún otro carácter no alfanumérico en el esquema o las políticas de ningún otro tipo de OIDC IdP.

Un esquema de notación de puntos anida cada instancia de un `:` carácter como elemento secundario de la frase `cognito` o frase `custom` inicial, como se muestra en el siguiente ejemplo:

```
"CognitoUser": {
   "shape": {
      "type": "Record",
      "attributes": {
         "cognito": {
            "type": "Record",
            "required": true,
            "attributes": {
               "username": {
                  "type": "String",
                  "required": true
               }
            }
         },
         "custom": {
            "type": "Record",
            "required": true,
            "attributes": {
               "employmentStoreCode": {
                  "type": "String",
                  "required": true
               }
            }
         },
         "email": {
            "type": "String"
         },
         "tenant": {
            "type": "String",
            "required": true
         }
      }
   }
}
```

Para ver un ejemplo de política que se validará con este esquema y utilizará la notación de puntos, consulteUtiliza la notación de puntos para hacer referencia a los atributos.

Implementación de la autorización en Amazon Verified Permissions

Tras crear el almacén de políticas, las políticas, las plantillas, el esquema y el modelo de autorización, estará listo para empezar a autorizar las solicitudes mediante los permisos verificados de Amazon. Para implementar la autorización de permisos verificados, debe combinar la configuración de las políticas AWS con la integración en una aplicación. Para integrar los permisos verificados en su aplicación, añada AWS SDK e implemente los métodos que invocan los permisos verificados API y generan decisiones de autorización en su almacén de políticas.

La autorización con permisos verificados es útil para los permisos de experiencia de usuario y APIlos permisos de sus aplicaciones.

Permisos de UX

> Controle el acceso de los usuarios a la UX de su aplicación. Puede permitir que un usuario vea solo los formularios, botones, gráficos y otros recursos exactos a los que necesita acceder. Por ejemplo, cuando un usuario inicia sesión, es posible que desee determinar si el botón «Transferir fondos» está visible en su cuenta. También puedes controlar las acciones que puede realizar un usuario. Por ejemplo, en la misma aplicación bancaria, es posible que desee determinar si su usuario puede cambiar la categoría de una transacción.

APIpermisos

> Controle el acceso de los usuarios a los datos. Las aplicaciones suelen formar parte de un sistema distribuido y reciben información de fuentes externasAPIs. En el ejemplo de la aplicación bancaria en la que los permisos verificados permiten mostrar el botón «Transferir fondos», se debe tomar una decisión de autorización más compleja cuando el usuario inicia una transferencia. Los permisos verificados pueden autorizar la API solicitud en la que se indican las cuentas de destino que son destinatarios de la transferencia aptos y, a continuación, la solicitud para transferir la transferencia a la otra cuenta.

Los ejemplos que ilustran este contenido provienen de un ejemplo de almacén de políticas. Para continuar, cree el almacén de políticas de DigitalPetStoremuestra en su entorno de pruebas.

Para ver un ejemplo de aplicación integral que implementa permisos de experiencia de usuario mediante la autorización por lotes, consulte Uso de permisos verificados de Amazon para obtener una autorización detallada a gran escala en el AWS blog de seguridad.

Temas

Operaciones disponibles para API la autorización

Los permisos verificados API tienen las siguientes operaciones de autorización.

IsAuthorized

La `IsAuthorized` API operación es el punto de entrada a las solicitudes de autorización con permisos verificados. Debe enviar los elementos principales, de acción, de recursos, de contexto y de entidad. Los permisos verificados validan las entidades de su solicitud en función del esquema del almacén de políticas. A continuación, Verified Permissions evalúa la solicitud comparándola con todas las políticas del almacén de políticas solicitado que se aplican a las entidades de la solicitud.

IsAuthorizedWithToken

La `IsAuthorizedWithToken` operación genera una solicitud de autorización a partir de los datos de usuario de los JSON web tokens de Amazon Cognito ()JWTs. Verified Permissions funciona directamente con Amazon Cognito como fuente de identidad en su almacén de políticas. Verified Permissions rellena todos los atributos del principal de su solicitud a partir de las afirmaciones que figuran en la identificación de los usuarios o en los tokens de acceso. Puede autorizar acciones y recursos a partir de los atributos de usuario o la pertenencia a un grupo en un grupo de usuarios de Amazon Cognito.

No puede incluir información sobre los tipos principales de grupos o usuarios en una `IsAuthorizedWithToken` solicitud. Debe rellenar todos los datos principales con los JWT que haya proporcionado.

BatchIsAuthorized

La `BatchIsAuthorized` operación procesa varias decisiones de autorización para un único principal o recurso en una sola API solicitud. Esta operación agrupa las solicitudes en una sola operación por lotes que minimiza el uso de la cuota y devuelve las decisiones de autorización para cada una de las 30 acciones anidadas complejas. Con la autorización por lotes para un único recurso, puede filtrar las acciones que un usuario puede realizar en un recurso. Con la

autorización por lotes para un único principal, puede filtrar los recursos sobre los que un usuario puede realizar acciones.

BatchIsAuthorizedWithToken

La `BatchIsAuthorizedWithToken` operación procesa varias decisiones de autorización para un único principal en una sola API solicitud. El principal lo proporciona la fuente de identidad del almacén de políticas en un identificador o token de acceso. Esta operación agrupa las solicitudes en una sola operación por lotes que minimiza el uso de la cuota y devuelve las decisiones de autorización para cada una de las 30 solicitudes de acciones y recursos como máximo. En sus políticas, puede autorizar su acceso desde sus atributos o su pertenencia a un grupo de usuarios de Amazon Cognito.

Del mismo `IsAuthorizedWithToken` modo, no puede incluir información sobre los principales tipos de grupos o usuarios en una `BatchIsAuthorizedWithToken` solicitud. Debe rellenar todos los datos principales con los JWT que haya proporcionado.

Probar su modelo de autorización

Para comprender el efecto de la decisión de autorización de permisos verificados de Amazon al implementar su aplicación, puede evaluar sus políticas a medida que las desarrolla con los permisos verificados Uso del banco de pruebas de permisos verificados de Amazon y con HTTPS REST API las solicitudes de estos. El banco de pruebas es una herramienta que sirve AWS Management Console para evaluar las solicitudes de autorización y las respuestas en su almacén de políticas.

Los permisos verificados REST API son el siguiente paso en su desarrollo, a medida que pasa de la comprensión conceptual al diseño de la aplicación. Los permisos verificados API aceptan solicitudes de autorización con IsAuthorizedy BatchIsAuthorizedcomo AWS APIsolicitudes firmadas dirigidas a los puntos finales de servicio regionales. IsAuthorizedWithToken Para probar su modelo de autorización, puede generar solicitudes con cualquier API cliente y comprobar que sus políticas devuelven las decisiones de autorización según lo previsto.

Por ejemplo, puede realizar una prueba `IsAuthorized` en un almacén de políticas de muestra con el siguiente procedimiento.

Test bench

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Cree un almacén de políticas a partir del almacén de políticas de muestra con el nombre DigitalPetStore.
2. Selecciona Test bench en tu nuevo almacén de políticas.
3. Rellene el formulario de solicitud de banco de pruebas IsAuthorizeden la API referencia de permisos verificados. Los siguientes detalles reproducen las condiciones del ejemplo 4 que hace referencia a la DigitalPetStoremuestra.

    a. Pon a Alice como la directora. Para que el director tome medidas, elige `DigitalPetStore::User` e ingresa`Alice`.

    b. Establece el rol de Alice como cliente. Elija Agregar un padre`DigitalPetStore::Role`, elija e introduzca Cliente.

    c. Establezca el recurso como pedido «1234». En el caso del recurso sobre el que actúa el principal, selecciónelo `DigitalPetStore::Order` e introdúzcalo`1234`.

    d. El `DigitalPetStore::Order` recurso requiere un `owner` atributo. Establece a Alice como la propietaria del pedido. Elige `DigitalPetStore::User` e ingresa `Alice`

    e. Alice solicitó ver el pedido. Para la acción que está tomando el director, elija`DigitalPetStore::Action::"GetOrder"`.
4. Elija Ejecutar solicitud de autorización. En un almacén de políticas sin modificar, esta solicitud da lugar a una `ALLOW` decisión. Tenga en cuenta la política de satisfacción que dio lugar a la decisión.
5. Elija Políticas en la barra de navegación izquierda. Revise la política estática con la descripción Customer Role: Get Order.
6. Observe que los permisos verificados permitieron la solicitud porque el principal tenía un rol de cliente y era el propietario del recurso.

REST API

1. Abra la consola de permisos verificados en https://console.aws.amazon.com/verifiedpermissions/. Cree un almacén de políticas a partir del almacén de políticas de muestra con el nombre DigitalPetStore.
2. Anote el ID del almacén de políticas del nuevo almacén de políticas.

3. IsAuthorizedEn la API referencia de permisos verificados, copia el cuerpo de la solicitud del ejemplo 4 que hace referencia al DigitalPetStoreejemplo.
4. Abra su API cliente y cree una solicitud al punto final del servicio regional de su almacén de políticas. Rellene los encabezados como se muestra en el ejemplo.
5. Pegue el ejemplo del cuerpo de la solicitud y cambie el valor por el ID del almacén de `policyStoreId` políticas que indicó anteriormente.
6. Envía la solicitud y revisa los resultados. En un almacén DigitalPetStorede políticas predeterminado, esta solicitud devuelve una `ALLOW` decisión.

Puede realizar cambios en las políticas, el esquema y las solicitudes de su entorno de prueba para cambiar los resultados y tomar decisiones más complejas.

1. Cambia la solicitud de forma que modifique la decisión de Verified Permissions. Por ejemplo, cambia el rol de Alice a `Employee` o cambia el `owner` atributo del pedido 1234 a`Bob`.
2. Cambie las políticas de manera que afecten a las decisiones de autorización. Por ejemplo, modifique la política con la descripción Customer Role: Get Order para eliminar la condición de que `User` debe ser el propietario del pedido `Resource` y modifique la solicitud para que `Bob` desee ver el pedido.
3. Cambie el esquema para permitir que las políticas tomen una decisión más compleja. Actualice las entidades solicitadas para que Alice pueda cumplir con los nuevos requisitos. Por ejemplo, edite el esquema `User` para poder ser miembro de `ActiveUsers` o`InactiveUsers`. Actualice la política para que solo los usuarios activos puedan ver sus propios pedidos. Actualice las entidades de la solicitud para que Alice sea una usuaria activa o inactiva.

Integrar sus modelos de autorización con las aplicaciones

Para implementar los permisos verificados de Amazon en tu aplicación, debes definir las políticas y el esquema que deseas que aplique tu aplicación. Con el modelo de autorización establecido y probado, el siguiente paso es empezar a generar API solicitudes desde el punto de vista de su cumplimiento. Para ello, debe configurar la lógica de la aplicación para recopilar los datos de los usuarios y rellenarlos para las solicitudes de autorización.

Cómo autoriza una aplicación las solicitudes con permisos verificados

1. Recopila información sobre el usuario actual. Por lo general, los detalles de un usuario se proporcionan en los detalles de una sesión autenticada, como una JWT cookie de sesión

web. Estos datos de usuario pueden proceder de una fuente de identidad de Amazon Cognito vinculada a su almacén de políticas o de otro proveedor de OpenID Connect (). OIDC

2. Recopile información sobre el recurso al que quiere acceder un usuario. Por lo general, la aplicación recibirá información sobre el recurso cuando un usuario haga una selección que requiera que la aplicación cargue un nuevo activo.
3. Determina la acción que el usuario quiere realizar.
4. Genera una solicitud de autorización para Verified Permissions con el principal, la acción, el recurso y las entidades que el usuario intentó realizar la operación. Verified Permissions evalúa la solicitud comparándola con las políticas de tu almacén de políticas y devuelve una decisión de autorización.
5. La aplicación lee la respuesta de autorización o denegación de Verified Permissions y aplica la decisión sobre la solicitud del usuario.

APILas operaciones de permisos verificados están integradas. AWS SDKs Para incluir los permisos verificados en una aplicación, integre el AWS SDK idioma que elija en el paquete de la aplicación.

Para obtener más información y descargarla AWS SDKs, consulta Herramientas para Amazon Web Services.

Los siguientes son enlaces a la documentación de varios recursos sobre permisos verificados AWS SDKs.

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto)
- AWS SDK for Ruby

El siguiente AWS SDK for JavaScript ejemplo `IsAuthorized` se origina en Simplifique la autorización detallada con Amazon Verified Permissions y Amazon Cognito.

```
const authResult = await avp.isAuthorized({
```

```
    principal: 'User::"alice"',
    action: 'Action::"view"',
    resource: 'Photo::"VacationPhoto94.jpg"',
    // whenever our policy references attributes of the entity,
    // isAuthorized needs an entity argument that provides
    // those attributes
    entities: {
       entityList: [
         {
            "identifier": {
                "entityType": "User",
                "entityId": "alice"
            },
            "attributes": {
                "location": {
                    "String": "USA"
                }
            }
         }
       ]
    }
});
```

Más recursos para desarrolladores

- Taller sobre permisos verificados de Amazon
- Permisos verificados de Amazon - Recursos
- Implemente un proveedor de políticas de autorización personalizado paraASP. NETAplicaciones principales que utilizan permisos verificados de Amazon
- Cree un servicio de asignación de derechos para aplicaciones empresariales con Amazon Verified Permissions
- Simplifique la autorización detallada con Amazon Verified Permissions y Amazon Cognito

Añadir contexto

El contexto es la información relevante para las decisiones políticas, pero no forma parte de la identidad del director, la acción o el recurso. Es posible que desee permitir una acción solo desde un conjunto de direcciones IP de origen o solo si el usuario ha iniciado sesión con MFA. Su aplicación tiene acceso a estos datos de sesión contextuales y debe rellenarlos para las solicitudes de autorización. Los datos de contexto de una solicitud de autorización de permisos verificados deben tener formato JSON en un elemento. `contextMap`

Los ejemplos que ilustran este contenido provienen de un almacén de políticas de muestra. Para continuar, cree el almacén de políticas de DigitalPetStoremuestra en su entorno de pruebas.

El siguiente objeto de contexto declara uno de cada tipo de datos de Cedar para una aplicación según el almacén de DigitalPetStorepolíticas de muestra.

```
"context": {
   "contextMap": {
      "MfaAuthorized": {
         "boolean": true
      },
      "AccountCodes": {
         "set": [
            {
               "long": 111122223333
            },
            {
               "long": 444455556666
            },
            {
               "long": 123456789012
            }
         ]
      },
      "UserAgent": {
         "string": "My UserAgent 1.12"
      },
      "RequestedOrderCount": {
         "long": 4
      },
      "NetworkInfo": {
         "record": {
```

```
            "IPAddress": {
               "string": "192.0.2.178"
            },
            "Country": {
               "string": "United States of America"
            },
            "SSL": {
               "boolean": true
            }
         }
      },
      "approvedBy": {
         "entityIdentifier": {
            "entityId": "Bob",
            "entityType": "DigitalPetStore::User"
         }
      }
   }
}
```

Tipos de datos en el contexto de la autorización

Booleano

Un binario `true` o un `false` valor. En el ejemplo, el valor booleano `true` for `MfaAuthenticated` indica que el cliente ha realizado una autenticación multifactorial antes de solicitar ver su pedido.

Establezca

Colección de elementos contextuales. Los miembros del conjunto pueden ser todos del mismo tipo, como en este ejemplo, o de tipos diferentes, incluido un conjunto anidado. En el ejemplo, el cliente está asociado a 3 cuentas diferentes.

Cadena

Secuencia de letras, números o símbolos, encerrados entre " caracteres. En el ejemplo, la `UserAgent` cadena representa el navegador que el cliente utilizó para solicitar ver su pedido.

Largo

Un número entero. En el ejemplo, `RequestedOrderCount` indica que esta solicitud forma parte de un lote que surgió cuando el cliente solicitó ver cuatro de sus pedidos anteriores.

Registro

Colección de atributos. Debe declarar estos atributos en el contexto de la solicitud. Un almacén de políticas con un esquema debe incluir esta entidad y los atributos de la entidad en el esquema. En el ejemplo, el `NetworkInfo` registro contiene información sobre la IP de origen del usuario, la geolocalización de esa IP determinada por el cliente y el cifrado en tránsito.

EntityIdentifier

Una referencia a una entidad y a los atributos declarados en el `entities` elemento de la solicitud. En el ejemplo, el empleado aprobó el pedido del usuario`Bob`.

Para probar este contexto de ejemplo en la DigitalPetStoreaplicación de ejemplo, debes actualizar tu solicitud`entities`, el esquema del almacén de políticas y la política estática con la descripción Customer Role: Get Order.

Modificar DigitalPetStore para aceptar el contexto de autorización

Inicialmente, no DigitalPetStorees un almacén de políticas muy complejo. No incluye políticas ni atributos de contexto preconfigurados para respaldar el contexto que hemos presentado. Para evaluar un ejemplo de solicitud de autorización con esta información contextual, realice las siguientes modificaciones en su almacén de políticas y en su solicitud de autorización.

Schema

Aplique las siguientes actualizaciones al esquema del almacén de políticas para admitir los nuevos atributos de contexto. Actualice `GetOrder` de `actions` la siguiente manera.

```
"GetOrder": {
   "memberOf": [],
   "appliesTo": {
      "resourceTypes": [
         "Order"
      ],
      "context": {
         "type": "Record",
         "attributes": {
            "UserAgent": {
               "required": true,
               "type": "String"
            },
```

```
            "approvedBy": {
               "name": "User",
               "required": true,
               "type": "Entity"
            },
            "AccountCodes": {
               "type": "Set",
               "required": true,
               "element": {
                  "type": "Long"
               }
            },
            "RequestedOrderCount": {
               "type": "Long",
               "required": true
            },
            "MfaAuthorized": {
               "type": "Boolean",
               "required": true
            }
         }
      },
      "principalTypes": [
         "User"
      ]
   }
}
```

Para hacer referencia al tipo de `record` datos mencionado `NetworkInfo` en el contexto de la solicitud, cree una construcción CommonType en el esquema de la siguiente manera. Una `commonType` construcción es un conjunto compartido de atributos que se pueden aplicar a distintas entidades.

> **Note**
>
> El editor de esquemas visuales de permisos verificados actualmente no admite `commonType` construcciones. Al añadirlos al esquema, ya no podrá ver el esquema en modo visual.

```
"commonTypes": {
```

```
   "NetworkInfo": {
      "attributes": {
         "IPAddress": {
            "type": "String",
            "required": true
         },
         "SSL": {
            "required": true,
            "type": "Boolean"
         },
         "Country": {
            "required": true,
            "type": "String"
         }
      },
      "type": "Record"
   }
}
```

Policy

La siguiente política establece las condiciones que debe cumplir cada uno de los elementos de contexto proporcionados. Se basa en la política estática existente con la descripción Customer Role: Get Order. Inicialmente, esta política solo requiere que el principal que realiza una solicitud sea el propietario del recurso.

```
permit (
    principal in DigitalPetStore::Role::"Customer",
    action in [DigitalPetStore::Action::"GetOrder"],
    resource
) when {
    principal == resource.owner &&
    context.MfaAuthorized == true &&
    context.UserAgent like "*My UserAgent*" &&
    context.RequestedOrderCount <= 4 &&
    context.AccountCodes.contains(111122223333) &&
    context.NetworkInfo.Country like "*United States*" &&
    context.NetworkInfo.SSL == true &&
    context.NetworkInfo.IPAddress like "192.0.2.*" &&
    context.approvedBy in DigitalPetStore::Role::"Employee"
};
```

Ahora exigimos que la solicitud de recuperación de un pedido cumpla con las condiciones de contexto adicionales que añadimos a la solicitud.

1. El usuario debe haber iniciado sesión con MFA.
2. El navegador web del usuario `User-Agent` debe contener la cadena`My UserAgent`.
3. El usuario debe haber solicitado ver 4 pedidos o menos.
4. Uno de los códigos de cuenta del usuario debe ser`111122223333`.
5. La dirección IP del usuario debe tener su origen en los Estados Unidos, debe estar en una sesión cifrada y su dirección IP debe empezar por`192.0.2.`.
6. Un empleado debe haber aprobado su pedido. En el `entities` elemento de la solicitud de autorización, declararemos un usuario Bob que tiene la función de`Employee`.

Request body

Tras configurar el almacén de políticas con el esquema y la política adecuados, puede presentar esta solicitud de autorización a la operación de la API de permisos verificados IsAuthorized. Tenga en cuenta que el `entities` segmento contiene una definición de Bob un usuario con un rol de`Employee`.

```
{
   "principal": {
      "entityType": "DigitalPetStore::User",
      "entityId": "Alice"
   },
   "action": {
      "actionType": "DigitalPetStore::Action",
      "actionId": "GetOrder"
   },
   "resource": {
      "entityType": "DigitalPetStore::Order",
      "entityId": "1234"
   },
   "context": {
     "contextMap": {
      "MfaAuthorized": {
       "boolean": true
      },
      "UserAgent": {
       "string": "My UserAgent 1.12"
```

```
      },
      "RequestedOrderCount":{
       "long": 4
      },
      "AccountCodes": {
       "set": [
        {"long": 111122223333},
        {"long": 444455556666},
        {"long": 123456789012}
        ]
      },
      "NetworkInfo": {
        "record": {
          "IPAddress": {"string": "192.0.2.178"},
          "Country": {"string": "United States of America"},
          "SSL": {"boolean": true}
        }
      },
      "approvedBy": {
        "entityIdentifier": {
         "entityId": "Bob",
         "entityType": "DigitalPetStore::User"
        }
      }
     }
   },
   "entities": {
     "entityList": [
       {
         "identifier": {
            "entityType": "DigitalPetStore::User",
            "entityId": "Alice"
         },
         "attributes": {
            "memberId": {
              "string": "801b87f2-1a5c-40b3-b580-eacad506d4e6"
              }
         },
         "parents": [
            {
            "entityType": "DigitalPetStore::Role",
            "entityId": "Customer"
            }
         ]
```

```
        },
        {
          "identifier": {
             "entityType": "DigitalPetStore::User",
             "entityId": "Bob"
          },
          "attributes": {
             "memberId": {
               "string": "49d9b81e-735d-429c-989d-93bec0bcfd8b"
               }
          },
          "parents": [
             {
             "entityType": "DigitalPetStore::Role",
             "entityId": "Employee"
             }
          ]
        },
        {
          "identifier": {
             "entityType": "DigitalPetStore::Order",
             "entityId": "1234"
          },
          "attributes": {
             "owner": {
               "entityIdentifier": {
                 "entityType": "DigitalPetStore::User",
                 "entityId": "Alice"
               }
             }
          },
          "parents": []
        }
      ]
    },
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Seguridad en Amazon Verified Permissions

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted 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 Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de conformidad que se aplican a los permisos verificados de Amazon, consulte AWS Servicios incluidos en el ámbito del programa de conformidad AWS .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted 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 le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Verified Permissions. En los siguientes temas, se le mostrará cómo configurar Verified Permissions 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 permisos verificados.

Temas

Protección de los datos en Amazon Verified Permissions

El modelo de se aplica a protección de datos en Amazon Verified Permissions. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la

Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Este contenido incluye las tareas de configuración y administración de la seguridad Servicios de AWS que utilizas. Para obtener más información sobre la privacidad de los datos, consulte la sección Privacidad de datos FAQ. Para obtener información sobre la protección de datos en Europa, consulte el modelo de responsabilidad AWS compartida y la entrada del GDPR blog sobre AWS seguridad.

- Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales.
- Recomendamos que proteja sus datos de las siguientes formas:
  - Utilice la autenticación multifactorial (MFA) con cada cuenta.
  - UseSSL/TLSpara comunicarse con AWS los recursos. Necesitamos TLS 1.2.
  - Configure API y registre la actividad del usuario con AWS CloudTrail.
  - Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
  - Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
  - Si necesita entre FIPS 140 y 2 módulos criptográficos validados para acceder a AWS través de una interfaz de línea de comandos o unaAPI, utilice un FIPS terminal. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la Norma federal de procesamiento de información () FIPS 140-2.
- Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con permisos verificados u otros tipos de permisos Servicios de AWS mediante la consola, API AWS CLI, o. AWS SDKs Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya información sobre las credenciales URL para validar la solicitud a ese servidor.
- Los nombres de sus acciones no deben incluir información confidencial.
- También le recomendamos encarecidamente que utilice siempre identificadores únicos, no mutables ni reutilizables para sus entidades (recursos y entidades principales). En un entorno de prueba, puede optar por utilizar identificadores de entidad simples, como `jane` o bob para el

nombre de una entidad de tipo `User`. Sin embargo, en un sistema de producción, es fundamental, por motivos de seguridad, utilizar valores únicos que no se puedan reutilizar. Le recomendamos que utilice valores como los identificadores únicos universales (UUIDs). Por ejemplo, piense en el usuario `jane` que deja la empresa. Más adelante, deja que otra persona use el nombre `jane`. Ese nuevo usuario tiene acceso automáticamente a todo lo que otorgan las políticas que aún hacen referencia a `User::"jane"`. Verified Permissions y Cedar no pueden distinguir entre el usuario nuevo y el anterior.

Esta directriz se aplica a los identificadores de las entidades principales y de los recursos. Utilice siempre identificadores con garantías de que son únicos y que no se han reutilizado nunca para asegurarse de no conceder acceso involuntariamente debido a la presencia de un identificador antiguo en una política.

- Asegúrese de que las cadenas que proporciona para definir los valores `Long` y `Decimal` estén dentro del rango válido de cada tipo. Además, asegúrese de que el uso de cualquier operador aritmético no dé como resultado un valor fuera del rango válido. Si se supera el rango, la operación produce una excepción de desbordamiento. Se omite una política que dé lugar a un error, lo que significa que una política de permisos podría no permitir el acceso de forma inesperada o que una política de prohibición podría no bloquear el acceso de forma inesperada.

Cifrado de datos

Amazon Verified Permissions cifra automáticamente todos los datos de los clientes, como las políticas, con una Clave administrada de AWS, por lo que no es necesario ni se admite el uso de una clave gestionada por el cliente.

Administración de identidades y accesos de Amazon Verified Permissions

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. IAM los administradores controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos de permisos verificados. IAM es uno Servicio de AWS que puede utilizar sin coste adicional.

Temas

- Autenticación con identidades
- Administración de acceso mediante políticas
- Cómo funciona Amazon Verified Permissions con IAM
- IAM políticas de permisos verificados
- Ejemplos de políticas basadas en identidad para Amazon Verified Permissions
- Solución de problemas de identidad y acceso de Amazon Verified Permissions

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realices en Verified Permissions.

Usuario de servicio: si utiliza el servicio Verified Permissions para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Verified Permissions para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en Verified Permissions, consulte Solución de problemas de identidad y acceso de Amazon Verified Permissions.

Administrador de servicio: si está a cargo de los recursos de Verified Permissions en su empresa, es probable que tenga acceso completo a Verified Permissions. Su trabajo consiste en determinar a qué características y recursos de Verified Permissions deben acceder los usuarios del servicio. A continuación, debe enviar solicitudes a su IAM administrador para cambiar los permisos de los usuarios del servicio. Revise la información de esta página para comprender los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM los permisos verificados, consulteCómo funciona Amazon Verified Permissions con IAM.

IAM administrador: si es IAM administrador, puede que desee obtener más información sobre cómo puede redactar políticas para administrar el acceso a los permisos verificados. Para ver ejemplos de políticas de permisos verificados basadas en la identidad que puede utilizar IAM, consulte. Ejemplos de políticas basadas en identidad para Amazon Verified Permissions

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como IAM usuario o asumiendo un IAM rol. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAMIdentity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, el administrador configuró previamente la federación de identidades mediante roles. IAM Cuando accede AWS mediante la federación, asume indirectamente un rol.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS incluye un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar AWS API las solicitudes en la Guía del IAM usuario.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactorial en la Guía del AWS IAM Identity Center usuario y Uso de la autenticación multifactorial (MFA) AWS en la Guía del IAM usuario.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para obtener la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM .

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones Cuentas de AWS . Para obtener información sobre IAM Identity Center, consulte ¿Qué es IAM Identity Center? en la Guía AWS IAM Identity Center del usuario.

Usuarios y grupos de IAM

Un IAMusuario es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM .

Un IAM grupo es una identidad que especifica un conjunto de IAM usuarios. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese Administradores de IAM y conceder permisos a dicho grupo para administrar los recursos de IAM .

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Cuándo crear un IAM usuario (en lugar de un rol) en la Guía del IAM usuario.

IAM roles

Un IAM rol es una identidad dentro de ti Cuenta de AWS que tiene permisos específicos. Es similar a un IAM usuario, pero no está asociado a una persona específica. Puede asumir temporalmente un IAM rol en el AWS Management Console cambiando de rol. Puede asumir un rol llamando a

una AWS API operación AWS CLI o o utilizando una operación personalizadaURL. Para obtener más información sobre los métodos de uso de roles, consulte Uso de IAM roles en la Guía del IAM usuario.

IAM los roles con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de los roles de federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM . Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en. IAM Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos IAM de usuario temporales: un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta obtenga acceso a los recursos de su cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso entre cuentas, consulte En qué se diferencian las IAM funciones de las políticas basadas en recursos en la Guía del usuario.IAM
- Aplicaciones en ejecución Amazon EC2: puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y que realizan o solicitan. AWS CLI AWS API Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulta Cómo usar un IAM rol para conceder permisos a las aplicaciones que se ejecutan en Amazon EC2 instancias en la Guía del IAM usuario.

Para saber si se deben usar IAM roles o IAM usuarios, consulte Cuándo crear un IAM rol (en lugar de un usuario) en la Guía del IAM usuario.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como JSON documentos. Para obtener más información sobre la estructura y el contenido de los documentos de JSON políticas, consulte Descripción general de JSON las políticas en la Guía del IAM usuario.

Los administradores pueden usar AWS JSON las políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

IAM las políticas definen los permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de AWS Management Console AWS CLI, el o el AWS API.

Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario.IAM

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y funciones de su empresa. Cuenta de AWS Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para obtener más información acerca de cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM .

Políticas basadas en recursos

Las políticas basadas en recursos son documentos JSON de políticas que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puede usar políticas AWS administradas desde una política IAM basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3 AWS WAF y Amazon VPC son ejemplos de servicios compatiblesACLs. Para obtener más informaciónACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAMusuario o rol). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM .

- Políticas de control de servicios (SCPs): SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. SCPLimita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre OrganizationsSCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para 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 IAM usuario.

Cómo funciona Amazon Verified Permissions con IAM

Antes de administrar el acceso IAM a los permisos verificados, obtén información sobre las IAM funciones disponibles para su uso con los permisos verificados.

IAM funciones que puedes usar con Amazon Verified Permissions

IAM función	Compatibilidad con Verified Permissions
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí

IAM función	Compatibilidad con Verified Permissions
Recursos de políticas	Sí
Claves de condición de política	No
ACLs	No
ABAC(etiquetas en las políticas)	No
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan los permisos verificados y otros AWS servicios con la mayoría de IAM las funciones, consulte AWS los servicios con los que funcionan IAM en la Guía del IAM usuario.

Políticas basadas en identidad para Verified Permissions

Compatibilidad con las políticas basadas en identidad	Sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario.IAM

Con las políticas IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para obtener más información sobre todos los elementos que

puede utilizar en una JSON política, consulte la referencia sobre los elementos de la IAM JSON política en la Guía del IAM usuario.

Ejemplos de políticas basadas en identidad para Verified Permissions

Para ver ejemplos de políticas basadas en identidad de Verified Permissions, consulte Ejemplos de políticas basadas en identidad para Amazon Verified Permissions.

Políticas basadas en recursos de Verified Permissions

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos JSON de política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar una cuenta completa o IAM entidades de otra cuenta como principales en una política basada en recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el IAM administrador de la cuenta de confianza también debe conceder permiso a la entidad principal (usuario o rol) para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte el tema Acceso a recursos entre cuentas IAM en la Guía del IAM usuario.

Acciones de política para Verified Permissions

Admite acciones de política	Sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de política suelen tener el mismo nombre que la AWS API operación asociada. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Verified Permissions, consulte Acciones definidas por Amazon Verified Permissions en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Verified Permissions utilizan el siguiente prefijo antes de la acción:

```
verifiedpermissions
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Get`, incluya la siguiente acción:

```
"Action": "verifiedpermissions:Get*"
```

Para ver ejemplos de políticas basadas en identidad de Verified Permissions, consulte Ejemplos de políticas basadas en identidad para Amazon Verified Permissions.

Recursos de políticas para Verified Permissions

Admite recursos de políticas	Sí

Los administradores pueden usar AWS JSON políticas 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` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso mediante su nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de recursos de permisos verificados y sus correspondientesARNs, consulte Tipos de recursos definidos por los permisos verificados de Amazon en la Referencia de autorización de servicio. Para saber con qué acciones puede especificar cada recurso, consulte Acciones definidas por Amazon Verified Permissions. ARN

Claves de condición de política de Amazon Verified Permissions

Admite claves de condición de políticas específicas del servicio	No

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder a un IAM usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de IAM usuario. Para obtener más información, consulte Elementos de la política de IAM : variables y etiquetas en la Guía del usuario de IAM .

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del IAM usuario.

ACLsen Permisos verificados

Soporta ACLs	No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

ABACcon permisos verificados

Soportes ABAC (etiquetas en las políticas)	No

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchos AWS recursos. Etiquetar entidades y recursos es el primer paso deABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABACes útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información al respectoABAC, consulte ¿Qué es? ABAC en la Guía IAM del usuario. Para ver un tutorial con los pasos de configuraciónABAC, consulte Usar el control de acceso basado en atributos (ABAC) en la Guía del IAM usuario.

Uso de credenciales temporales con Verified Permissions

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta la sección Servicios de AWS Cómo trabajar con credenciales temporales IAM en la Guía del IAM usuario.

Está utilizando credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información acerca del cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM .

Puede crear credenciales temporales manualmente con la AWS CLI tecla o. AWS API A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios de Verified Permissions

Admite permisos de entidades principales	Sí

Cuando utilizas un IAM usuario o un rol para realizar acciones en AWSél, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del principal que llama a una Servicio de AWS, junto con los que solicita, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones

con otros recursos Servicios de AWS o para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar las sesiones de acceso.

Roles de servicio para Verified Permissions

Compatible con roles de servicio	No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentro IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la IAM Guía del usuario.

Permisos de roles vinculados al servicio para Verified Permissions

Compatible con roles vinculados al servicio	No

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.

Para obtener más información sobre la creación o la administración de funciones vinculadas a un servicio, consulte los AWS servicios que funcionan con. IAM Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

IAM políticas de permisos verificados

Verified Permissions administra los permisos de los usuarios dentro de la aplicación. Para que su aplicación pueda acceder a los permisos verificados APIs o para que AWS Management Console los usuarios puedan gestionar las políticas de Cedar en un almacén de políticas de permisos verificados, debe añadir los IAM permisos necesarios.

Las políticas basadas en la identidad son documentos de política de JSON permisos que puede adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué

condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario. IAM

Con las políticas IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones (enumeradas a continuación). No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la referencia sobre los elementos de la IAM JSON política en la Guía del IAM usuario.

Action	Descripción
CreatePolicyStore	Acción para crear un nuevo almacén de políticas.
DeletePolicyStore	Acción para eliminar un almacén de políticas.
ListPolicyStores	Acción para enumerar todos los almacenes de pólizas del Cuenta de AWS.
CreatePolicy	Acción para crear una política de Cedar en un almacén de políticas. Puede crear una política estática o una política vinculada a una plantilla de política.
DeletePolicy	Acción para eliminar una política de un almacén de políticas.
GetPolicy	Acción para recuperar información sobre una política específica.
ListPolicies	Acción para enumerar todas las políticas de un almacén de políticas.
IsAuthorized	Acción para obtener una respuesta de autorización en función de los parámetros descritos en la solicitud de autorización.

Ejemplo IAM de política de permiso para la CreatePolicy acción:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "verifiedpermissions:CreatePolicy"
            ],
            "Resource": "*"
        }
    ]
 }
```

Ejemplos de políticas basadas en identidad para Amazon Verified Permissions

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de Verified Permissions. Tampoco pueden realizar tareas con las teclas AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS API. IAM El administrador debe crear IAM políticas que concedan permiso a los usuarios y a los roles para realizar acciones en los recursos que necesiten. El administrador debe asociar esas políticas a los usuarios que las necesiten.

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte Creación de IAM políticas en la Guía del IAM usuario.

Para obtener más información sobre las acciones y los tipos de recursos definidos por los permisos verificados, incluido el ARNs formato de cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de los permisos verificados de Amazon en la Referencia de autorización de servicios.

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la consola de Verified Permissions
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, acceder o eliminar los recursos de Verified Permissions en su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos en muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM .
- Aplique permisos con privilegios mínimos: cuando establezca permisos con IAM políticas, 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. Para obtener más información sobre cómo IAM aplicar permisos, consulte Políticas y permisos IAM en la IAM Guía del usuario.
- Utilice las condiciones en IAM las políticas para restringir aún más el acceso: puede añadir una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse medianteSSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte los elementos IAM JSON de la política: Condición en la Guía del IAM usuario.
- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAMAccess Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarle a crear políticas seguras y funcionales. Para obtener más información, consulte la validación de políticas de IAM Access Analyzer en la Guía del IAM usuario.
- Requerir autenticación multifactorial (MFA): si se encuentra en una situación en la que se requieren IAM usuarios o un usuario raíz Cuenta de AWS, actívela MFA para aumentar la seguridad. Para solicitarlo MFA cuando se convoque a API las operaciones, añada MFA

condiciones a sus políticas. Para obtener más información, consulte Configuración del API acceso MFA protegido en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadas IAM, consulte las prácticas recomendadas de seguridad IAM en la Guía del IAM usuario.

Uso de la consola de Verified Permissions

Para acceder a la consola de Amazon Verified Permissions, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de permisos verificados de 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 es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas al AWS CLI o al AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de permisos verificados, adjunte también la política de permisos verificados *ConsoleAccess* o *ReadOnly* AWS gestionada 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 se puede crear una política que permita a IAM los usuarios ver las políticas integradas y administradas asociadas a su identidad de usuario. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la tecla o. AWS CLI AWS API

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

Solución de problemas de identidad y acceso de Amazon Verified Permissions

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con Verified Permissions e IAM.

Temas

- No tengo autorización para realizar una acción en Verified Permissions
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de permisos verificados

No tengo autorización para realizar una acción en Verified Permissions

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

El siguiente ejemplo de error se produce cuando el `mateojackson` IAM usuario intenta usar la consola para ver los detalles de un *`my-example-widget`* recurso ficticio pero no tiene los `verifiedpermissions:`*`GetWidget`* permisos ficticios.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 verifiedpermissions: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 `verifiedpermissions:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con AWS el administrador. 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`, se deben actualizar las políticas a fin de permitirle pasar un rol a Verified Permissions.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un IAM usuario denominado `marymajor` intenta utilizar la consola para realizar una acción en Verified Permissions. 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 AWS el administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de permisos verificados

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para

que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para más información, consulte lo siguiente:

- Para obtener información acerca de si Verified Permissions admite estas características, consulte Cómo funciona Amazon Verified Permissions con IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a través de los Cuentas de AWS que eres propietario, consulta Cómo proporcionar acceso a un IAM usuario en otro de tu Cuenta de AWS propiedad en la Guía del IAM usuario.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo permitir el acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del IAM usuario.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticadoPara que su aplicación pueda acceder a s externamente (federación de identidades) en la Guía del usuario de IAM .
- Para saber la diferencia entre usar roles y políticas basadas en recursos para el acceso entre cuentas, consulta el tema Acceso a recursos entre cuentas IAM en la Guía del IAM usuario.

Validación de conformidad de Amazon Verified Permissions

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.

- Diseñar una arquitectura basada en HIPAA la seguridad y el cumplimiento Amazon Web Services: en este documento técnico se describe cómo pueden utilizar las empresas AWS para crear HIPAA aplicaciones aptas para ello.

> **Note**
>
> No todos son aptos. Servicios de AWS HIPAA Para obtener más información, consulta la Referencia de servicios HIPAA aptos.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. En las guías se resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y se orientan a los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, por ejemplo PCIDSS, cumpliendo con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS consumo para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia de Amazon Verified Permissions

La infraestructura AWS global se basa en Regiones de AWS zonas de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. 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.

Al crear un almacén de políticas de permisos verificados, se crea dentro de una persona Región de AWS y se replica automáticamente en los centros de datos que componen las zonas de disponibilidad de esa región. En este momento, Verified Permissions no admite ninguna replicación entre regiones.

Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura AWS global.

Supervisión de las API llamadas de Amazon Verified Permissions

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de Amazon Verified Permissions y del resto de sus AWS soluciones. AWS proporciona las siguientes herramientas para supervisar los permisos verificados, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario:

- AWS CloudTrailcaptura API las llamadas y los eventos relacionados realizados por su AWS cuenta o en su nombre y envía los archivos de registro a un bucket de Amazon S3 que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron. Para obtener más información, consulte la AWS CloudTrail Guía del usuario de .

Para obtener más información sobre cómo supervisar los permisos verificados con CloudTrail, consulteRegistrar las API llamadas de Amazon Verified Permissions mediante AWS CloudTrail.

Registrar las API llamadas de Amazon Verified Permissions mediante AWS CloudTrail

Amazon Verified Permissions 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 Verified Permissions. CloudTrail captura todas las API solicitudes de permisos verificados como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de permisos verificados y las llamadas en código a las API operaciones de permisos verificados. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para permisos verificados. 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, puedes determinar la solicitud que se realizó a Verified Permissions, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía AWS CloudTrail del usuario.

Información sobre permisos verificados en CloudTrail

CloudTrail está habilitada en tu cuenta Cuenta de AWS al crear la cuenta. Cuando se produce una actividad en los permisos verificados, 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 eventos recientes en su Cuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de tus eventos Cuenta de AWS, incluidos los eventos relacionados con los permisos verificados, crea un registro. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. 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 más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configuración de SNS las notificaciones de Amazon para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas las acciones de permisos verificados se registran CloudTrail y se documentan en la Guía de API referencia de permisos verificados de Amazon. Por ejemplo, las llamadas a las `CreateIdentitySource` `ListPolicyStores` acciones y las llamadas generan entradas en los archivos de CloudTrail registro. `DeletePolicy`

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 credenciales de usuario root o AWS Identity and Access Management (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 CloudTrail userIdentity elemento.

Los eventos de datos como IsAuthorized, por ejemplo, no se registran de forma predeterminada cuando se crea un banco de datos de rutas o eventos. IsAuthorizedWithToken Para registrar CloudTrail los eventos de datos, debe añadir de forma explícita los recursos o tipos de recursos compatibles para los que desea recopilar la actividad. Para obtener más información, consulte Eventos de datos en la Guía del usuario de AWS CloudTrail .

Descripción de las entradas del archivo de registro de Verified Permissions

Un rastro es una configuración que permite la entrega de 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 API llamadas públicas, por lo que no aparecen en ningún orden específico.

Temas

- IsAuthorized
- BatchIsAuthorized
- CreatePolicyStore
- ListPolicyStores
- DeletePolicyStore
- PutSchema
- GetSchema
- CreatePolicyTemplate
- DeletePolicyTemplate
- CreatePolicy
- GetPolicy
- CreateIdentitySource
- GetIdentitySource
- ListIdentitySources
- DeleteIdentitySource

> **Note**
>
> Algunos campos se han eliminado de los ejemplos por motivos de privacidad de datos.

IsAuthorized


```
{
    "eventVersion": "1.08",
    "userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLE_PRINCIPAL_ID",
  "arn": "arn:aws:iam::123456789012:role/ExampleRole",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2023-11-20T22:55:03Z",
    "eventSource": "verifiedpermissions.amazonaws.com",
    "eventName": "IsAuthorized",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
    "requestParameters": {
        "principal": {
            "entityType": "PhotoFlash::User",
            "entityId": "alice"
        },
        "action": {
            "actionType": "PhotoFlash::Action",
            "actionId": "ViewPhoto"
        },
        "resource": {
            "entityType": "PhotoFlash::Photo",
            "entityId": "VacationPhoto94.jpg"
        },
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "decision": "ALLOW"
    },
    "requestID": "346c4b6a-d12f-46b6-bc06-6c857bd3b28e",
```

```
    "eventID": "8a4fed32-9605-45dd-a09a-5ebbf0715bbc",
    "readOnly": true,
    "resources": [
        {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
}
```

BatchIsAuthorized

```
{
    "eventVersion": "1.08",
    "userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLE_PRINCIPAL_ID",
  "arn": "arn:aws:iam::123456789012:role/ExampleRole",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2023-11-20T23:02:33Z",
    "eventSource": "verifiedpermissions.amazonaws.com",
    "eventName": "BatchIsAuthorized",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-cli/2.11.18 Python/3.11.3 Linux/5.4.241-160.348.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/verifiedpermissions.is-authorized",
    "requestParameters": {
        "requests": [
            {
                "principal": {
                    "entityType": "PhotoFlash::User",
                    "entityId": "alice"
                },
                "action": {
                    "actionType": "PhotoFlash::Action",
```

```
                    "actionId": "ViewPhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            },
            {
                "principal": {
                    "entityType": "PhotoFlash::User",
                    "entityId": "annalisa"
                },
                "action": {
                    "actionType": "PhotoFlash::Action",
                    "actionId": "DeletePhoto"
                },
                "resource": {
                    "entityType": "PhotoFlash::Photo",
                    "entityId": "VacationPhoto94.jpg"
                }
            }
        ],
        "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    "responseElements": null,
    "additionalEventData": {
        "results": [
            {
                "request": {
                    "principal": {
                        "entityType": "PhotoFlash::User",
                        "entityId": "alice"
                    },
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "ViewPhoto"
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
                "decision": "ALLOW"
            },
```

```
            {
                "request": {
                    "principal": {
                        "entityType": "PhotoFlash::User",
                        "entityId": "annalisa"
                    },
                    "action": {
                        "actionType": "PhotoFlash::Action",
                        "actionId": "DeletePhoto"
                    },
                    "resource": {
                        "entityType": "PhotoFlash::Photo",
                        "entityId": "VacationPhoto94.jpg"
                    }
                },
                "decision": "DENY"
            }
        ]
    },
    "requestID": "a8a5caf3-78bd-4139-924c-7101a8339c3b",
    "eventID": "7d81232f-f3d1-4102-b9c9-15157c70487b",
    "readOnly": true,
    "resources": [
        {
    "accountId": "123456789012",
    "type": "AWS::VerifiedPermissions::PolicyStore",
    "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
}
```

CreatePolicyStore

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
```

```
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "validationSettings": {
      "mode": "OFF"
    }
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
    "createdDate": "2023-05-22T07:43:33.962794Z",
    "lastUpdatedDate": "2023-05-22T07:43:33.962794Z"
  },
  "requestID": "1dd9360e-e2dc-4554-ab65-b46d2cf45c29",
  "eventID": "b6edaeee-3584-4b4e-a48e-311de46d7532",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```


ListPolicyStores


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
```

```
  "eventTime": "2023-05-22T07:43:33Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListPolicyStores",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "maxResults": 10
  },
  "responseElements": null,
  "requestID": "5ef238db-9f87-4f37-ab7b-6cf0ba5df891",
  "eventID": "b0430fb0-12c3-4cca-8d05-84c37f99c51f",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

DeletePolicyStore

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeletePolicyStore",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "1368e8f9-130d-45a5-b96d-99097ca3077f",
  "eventID": "ac482022-b2f6-4069-879a-dd509123d8d7",
  "readOnly": false,
```

```
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

PutSchema

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T12:58:57Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "PutSchema",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T12:58:57.513442Z",
    "namespaces": "[some_namespace]",
    "createdDate": "2023-05-16T12:58:57.513442Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "631fbfa1-a959-4988-b9f8-f1a43ff5df0d",
  "eventID": "7cd0c677-733f-4602-bc03-248bae581fe5",
  "readOnly": false,
```

```
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetSchema

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-25T01:12:07Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetSchema",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "a1f4d4cd-6156-480a-a9b8-e85a71dcc7c2",
  "eventID": "0b3b8e3d-155c-46f3-a303-7e9e8b5f606b",
  "readOnly": true,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::VerifiedPermissions::PolicyStore",
```

```
      "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "222222222222",
  "eventCategory": "Management"
}
```

CreatePolicyTemplate

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-16T13:00:24Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "lastUpdatedDate": "2023-05-16T13:00:23.444404Z",
    "createdDate": "2023-05-16T13:00:23.444404Z",
    "policyTemplateId": "PTEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
  },
  "requestID": "73953bda-af5e-4854-afe2-7660b492a6d0",
  "eventID": "7425de77-ed84-4f91-a4b9-b669181cc57b",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
```

```
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

DeletePolicyTemplate

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::222222222222:role/ExampleRole",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-25T01:11:48Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeletePolicyTemplate",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "5ff0f22e-6bbd-4b85-a400-4fb74aa05dc6",
  "eventID": "c0e0c689-369e-4e95-a9cd-8de113d47ffa",
  "readOnly": false,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "ARN": "arn:aws:verifiedpermissions::222222222222:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
```

```
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "222222222222",
  "eventCategory": "Management"
}
```

CreatePolicy

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:42:30Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreatePolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
        "entityType": "PhotoApp::Role",
        "entityId": "PhotoJudge"
    },
    "resource": {
        "entityType": "PhotoApp::Application",
        "entityId": "PhotoApp"
    },
    "lastUpdatedDate": "2023-05-22T07:42:30.70852Z",
    "createdDate": "2023-05-22T07:42:30.70852Z"
  },
  "requestID": "93ffa151-3841-4960-9af6-30a7f817ef93",
```

```
  "eventID": "30ab405f-3dff-43ff-8af9-f513829e8bde",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

GetPolicy

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-22T07:43:29Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetPolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyId": "SPEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "23022a9e-2f5c-4dac-b653-59e6987f2fac",
  "eventID": "9b4d5037-bafa-4d57-b197-f46af83fc684",
  "readOnly": true,
  "resources": [
    {
```

```
      "accountId": "123456789012",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

CreateIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-19T01:27:44Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "CreateIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "clientToken": "a1b2c3d4-e5f6-a1b2-c3d4-TOKEN1111111",
    "configuration": {
      "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:000011112222:us-east-1:userpool/us-
east-1_aaaaaaaaaa"
      }
    },
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "principalEntityType": "User"
  },
  "responseElements": {
    "createdDate": "2023-07-14T15:05:01.599534Z",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
```

```
    "lastUpdatedDate": "2023-07-14T15:05:01.599534Z",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
  "requestID": "afcc1e67-d5a4-4a9b-a74c-cdc2f719391c",
  "eventID": "f13a41dc-4496-4517-aeb8-a389eb379860",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

GetIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:31Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "GetIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "7a6ecf79-c489-4516-bb57-9ded970279c9",
```

```
  "eventID": "fa158e6c-f705-4a15-a731-2cdb4bd9a427",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

ListIdentitySources

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T20:05:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "ListIdentitySources",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "95d2a7bc-7e9a-4efe-918e-97e558aacaf7",
  "eventID": "d3dc53f6-1432-40c8-9d1d-b9eeb75c6193",
  "readOnly": true,
  "resources": [
    {
      "accountId": "333333333333",
```

```
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

DeleteIdentitySource

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::333333333333:role/ExampleRole",
    "accountId": "333333333333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
  },
  "eventTime": "2023-05-24T19:55:32Z",
  "eventSource": "verifiedpermissions.amazonaws.com",
  "eventName": "DeleteIdentitySource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "aws-sdk-rust/0.55.2 os/linux lang/rust/1.69.0",
  "requestParameters": {
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
  },
  "responseElements": null,
  "requestID": "d554d964-0957-4834-a421-c417bd293086",
  "eventID": "fe4d867c-88ee-4e5d-8d30-2fbc208c9260",
  "readOnly": false,
  "resources": [
    {
      "accountId": "333333333333",
      "type": "AWS::VerifiedPermissions::PolicyStore",
      "arn": "arn:aws:verifiedpermissions::333333333333:policy-store/
PSEXAMPLEabcdefg111111"
    }
```

```
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "333333333333",
  "eventCategory": "Management"
}
```

Creación de recursos de Amazon Verified Permissions con AWS CloudFormation

Amazon Verified Permissions está integrado con AWS CloudFormation un servicio que le ayuda a modelar y configurar sus AWS recursos para que pueda dedicar menos tiempo a crear y administrar sus recursos e infraestructura. Usted crea una plantilla que describe todos los AWS recursos que desea (como los almacenes de políticas) y AWS CloudFormation aprovisiona y configura esos recursos por usted.

Cuando la utilice AWS CloudFormation, podrá reutilizar la plantilla para configurar los recursos de permisos verificados de forma coherente y repetida. Describa sus recursos una vez y, a continuación, aprovisione los mismos recursos una y otra vez en varias Cuentas de AWS regiones.

> **Important**
>
> Amazon Cognito Identity no está disponible al mismo tiempo que los permisos verificados de Regiones de AWS Amazon. Si recibe un error AWS CloudFormation relacionado con Amazon Cognito Identity, por ejemplo, le recomendamos que cree el grupo de usuarios y el cliente de Amazon Cognito en la zona geográfica más cercana a donde esté disponible Región de AWS Amazon Cognito Identity. `Unrecognized resource types: AWS::Cognito::UserPool, AWS::Cognito::UserPoolClient` Utilice este grupo de usuarios recién creado al crear la fuente de identidad de Verified Permissions.

Permisos y plantillas verificados AWS CloudFormation

Para aprovisionar y configurar los recursos de Verified Permissions y sus servicios relacionados, debe entender las plantillas de AWS CloudFormation. Las plantillas son archivos de texto formateados en JSON oYAML. Estas plantillas describen los recursos que deseas aprovisionar en tus AWS CloudFormation pilas. Si no estás familiarizado con JSON ellasYAML, puedes usar AWS CloudFormation Designer para ayudarte a empezar con AWS CloudFormation las plantillas. Para obtener más información, consulta ¿Qué es AWS CloudFormation Designer? en la Guía AWS CloudFormation del usuario.

Verified Permissions permite crear fuentes de identidad, políticas, almacenes de políticas y plantillas de políticas en AWS CloudFormation. Para obtener más información, incluidos ejemplos JSON y

YAML plantillas de recursos de permisos verificados, consulte la referencia del tipo de recurso de permisos verificados de Amazon en la Guía del AWS CloudFormation usuario.

Constructos AWS CDK

AWS Cloud Development Kit (AWS CDK) Se trata de un marco de desarrollo de software de código abierto para definir la infraestructura de nube en el código y aprovisionarla mediante ella. AWS CloudFormation Se pueden usar construcciones, o componentes de nube reutilizables, para crear plantillas. AWS CloudFormation Luego, estas plantillas se pueden usar para implementar su infraestructura de nube.

Para obtener más información y descargar AWS CDKs, consulte AWS Cloud Development Kit.

Los siguientes son enlaces a la documentación de los AWS CDK recursos de permisos verificados, como las construcciones.

- Amazon Verified Permissions L2 Construct CDK

Más información sobre AWS CloudFormation

Para obtener más información AWS CloudFormation, consulte los siguientes recursos:

- AWS CloudFormation
- AWS CloudFormation Guía del usuario
- AWS CloudFormation APIReferencia
- AWS CloudFormation Guía del usuario de la interfaz de línea de comandos

Acceda a los permisos verificados de Amazon mediante AWS PrivateLink

Puedes usarlo AWS PrivateLink para crear una conexión privada entre tus permisos verificados VPC y los de Amazon. Puede acceder a los permisos verificados como si estuvieran en los suyosVPC, sin necesidad de utilizar una pasarela de Internet, un NAT dispositivo, una VPN conexión o una AWS Direct Connect conexión. Las instancias VPC que tengas no necesitan direcciones IP públicas para acceder a los permisos verificados.

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 Verified Permissions.

Para obtener más información, consulte Acceso a los Servicios de AWS a través de AWS PrivateLink en la Guía de AWS PrivateLink .

Consideraciones sobre Verified Permissions

Antes de configurar un punto de conexión de interfaz para Verified Permissions, consulte la sección Consideraciones en la Guía de AWS PrivateLink .

Verified Permissions permite realizar llamadas a todas sus API acciones a través del punto final de la interfaz.

VPCLos permisos verificados no admiten políticas de puntos finales. De forma predeterminada, el acceso completo a Verified Permissions se permite a través del punto de conexión. Como alternativa, puede asociar un grupo de seguridad a las interfaces de red de los puntos de conexión para controlar el tráfico a Verified Permissions a través del punto de conexión de interfaz.

Crear un punto de conexión de interfaz para Verified Permissions

Puedes crear un punto final de interfaz para los permisos verificados mediante la VPC consola de Amazon o con 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 de conexión para Verified Permissions utilizando el siguiente nombre de servicio:

```
com.amazonaws.region.verifiedpermissions
```

Si habilitas el modo privado DNS para el punto final de la interfaz, puedes realizar API solicitudes a los permisos verificados utilizando su DNS nombre regional predeterminado. Por ejemplo, `verifiedpermissions.us-east-1.amazonaws.com`.

Cuotas para Amazon Verified Permissions

Cuenta de AWS Tiene cuotas predeterminadas, anteriormente denominadas límites, para cada AWS servicio. A menos que se indique lo contrario, cada cuota es específica de la región de . Puede solicitar el aumento de algunas cuotas, pero otras no se pueden aumentar.

Para ver todas las cuotas de Verified Permissions, abra la consola de Service Quotas. En el panel de navegación, elija Servicios de AWS y seleccione Verified Permissions.

Para solicitar un aumento de cuota, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas. Si la cuota aún no se encuentra disponible en Service Quotas, utilice el formulario de aumento del límite.

Cuenta de AWS Tiene las siguientes cuotas relacionadas con los permisos verificados.

Temas

Cuotas de recursos

Nombre	Valor predeterminado	Ajusta	Descripción
Almacenes de políticas por región y por cuenta	Cada región admitida: 1000	Sí	El número máximo de almacenes de políticas.
Plantillas de política por almacén de políticas	Cada región admitida: 40	Sí	El número máximo de plantillas de política en un almacén de políticas.
Fuentes de identidad por almacén de políticas	1	No	El número máximo de fuentes de identidad que puede definir para un almacén de políticas.

Nombre	Valor predeterminado	Ajusta	Descripción
Tamaño de la solicitud de autorización[1]	1 MB	No	Tamaño máximo de una solicitud de autorización.
Tamaño de la póliza	10 000 bytes	No	El tamaño máximo de una política individual.
Tamaño del esquema	100 000 bytes	No	El tamaño máximo del esquema de un almacén de políticas.
Tamaño de la política por recurso	200 000 bytes[2]	No	El tamaño máximo de todas las políticas que hacen referencia a un recurso específico.

[1] La cuota para una solicitud de autorización es la misma para ambos IsAuthorized. IsAuthorizedWithToken

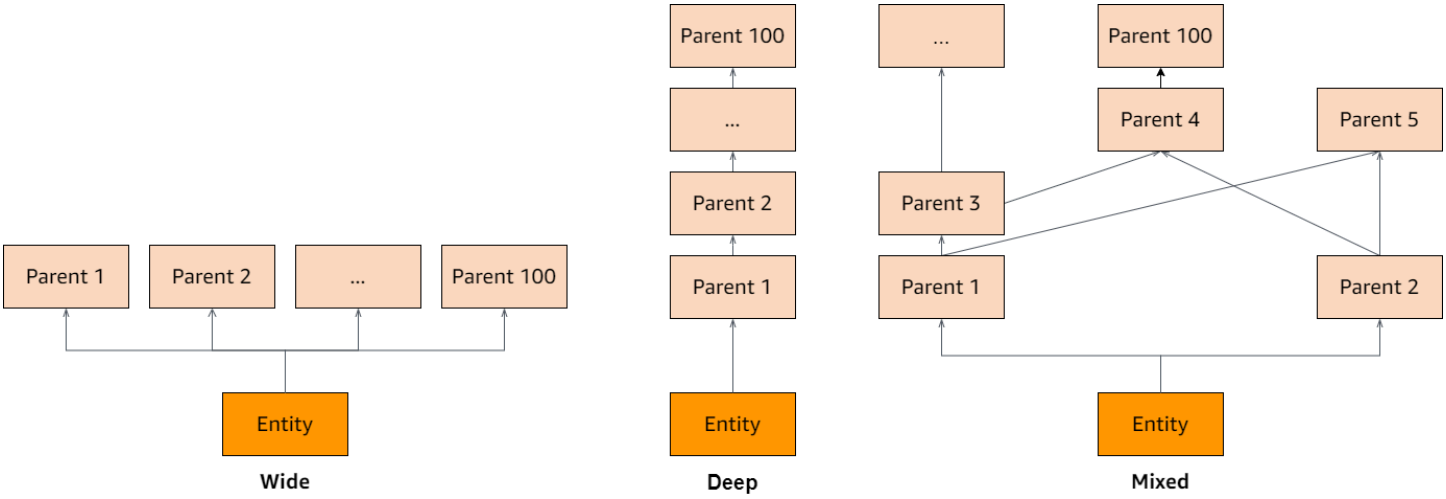
[2] El tamaño total de todas las políticas relacionadas con un único recurso no puede superar los 200 000 bytes. Además, el tamaño total de todas las políticas que especifican «Todos los recursos» no puede superar los 200 000 bytes. En el caso de las políticas vinculadas a plantillas, el tamaño de la plantilla de política se cuenta solo una vez, más el tamaño de cada conjunto de parámetros utilizado para crear una instancia de cada política vinculada a una plantilla.

Cuotas para jerarquías

Nombre	Valor predeterminado	Ajusta	Descripción
Elementos principales transitivos por entidad principal	100	No	El número máximo de elementos principales transitivos para cada entidad principal.

Nombre	Valor predeterminado	Ajusta	Descripción
Elementos principales transitivos por acción	100	No	El número máximo de elementos principales transitivos para cada acción.
Elementos principales transitivos por recurso	100	No	El número máximo de elementos principales transitivos para cada recurso.

El siguiente diagrama ilustra cómo se pueden definir los elementos principales transitivos para una entidad (principal, acción o recurso).



Cuotas de operaciones por segundo

Los permisos verificados limitan las solicitudes a los puntos finales del servicio Región de AWS cuando las solicitudes de aplicaciones superan la cuota de una operación. API Los permisos verificados pueden generar una excepción si superas la cuota de solicitudes por segundo o si intentas realizar operaciones de escritura simultáneas. Puede ver sus RPS cuotas actuales en Service Quotas. Para evitar que las aplicaciones superen la cuota de una operación, debe optimizarlas para que tengan en cuenta los reintentos y los retrasos exponenciales. Para obtener

más información, consulte Reintentar con un patrón de retroceso y Administrar y supervisar API la limitación de las cargas de trabajo.

Nombre	Valor predeterminado	Ajusta	Descripción
BatchIsAuthorized solicitudes por segundo, por región y por cuenta	Cada región admitida: 30	Sí	El número máximo de BatchIsAuthorized solicitudes por segundo.
BatchIsAuthorizedWithToken solicitudes por segundo por región y por cuenta	Cada región admitida: 30	Sí	El número máximo de BatchIsAuthorizedWithToken solicitudes por segundo.
CreatePolicy solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de CreatePolicy solicitudes por segundo.
CreatePolicyStore solicitudes por segundo por región y por cuenta	Cada región admitida: 1	No	El número máximo de CreatePolicyStore solicitudes por segundo.
CreatePolicyTemplate solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de CreatePolicyTemplate solicitudes por segundo.
DeletePolicy solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de DeletePolicy solicitudes por segundo.
DeletePolicyStore solicitudes por segundo por región y por cuenta	Cada región admitida: 1	No	El número máximo de DeletePolicyStore solicitudes por segundo.
DeletePolicyTemplate solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de DeletePolicyTemplate solicitudes por segundo.

Nombre	Valor predeterminado	Ajusta	Descripción
GetPolicy solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de GetPolicy solicitudes por segundo.
GetPolicyTemplate solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de GetPolicyTemplate solicitudes por segundo.
GetSchema solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de GetSchema solicitudes por segundo.
IsAuthorized solicitudes por segundo por región y por cuenta	Cada región admitida: 200	Sí	El número máximo de IsAuthorized solicitudes por segundo.
IsAuthorizedWithToken solicitudes por segundo por región y por cuenta	Cada región admitida: 200	Sí	El número máximo de IsAuthorizedWithToken solicitudes por segundo.
ListPolicies solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de ListPolicies solicitudes por segundo.
ListPolicyStores solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de ListPolicyStores solicitudes por segundo.
ListPolicyTemplates solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de ListPolicyTemplates solicitudes por segundo.
PutSchema solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de PutSchema solicitudes por segundo.

Nombre	Valor predeterminado	Ajusta	Descripción
UpdatePolicy solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de UpdatePolicy solicitudes por segundo.
UpdatePolicyStore solicitudes por segundo por región y por cuenta	Cada región admitida: 10	No	El número máximo de UpdatePolicyStore solicitudes por segundo.
UpdatePolicyTemplate solicitudes por segundo por región y por cuenta	Cada región admitida: 10	Sí	El número máximo de UpdatePolicyTemplate solicitudes por segundo.

Términos y conceptos lingüísticos de la política de permisos verificados de Amazon y Cedar

Debe comprender los siguientes conceptos para utilizar Amazon Verified Permissions.

Conceptos de Verified Permissions

- Modelo de autorización
- Solicitud de autorización
- Respuesta de autorización
- Políticas consideradas
- Datos de contexto
- Políticas determinantes
- Datos de la entidad
- Permisos, autorizaciones y entidades principales
- Aplicación de políticas
- Almacén de políticas
- Políticas satisfechas
- Diferencias entre los permisos verificados de Amazon y el lenguaje de la política de Cedar

Conceptos del lenguaje de políticas de Cedar

- Autorización
- Entidad
- Grupos y jerarquías
- Espacios de nombres
- Política
- Plantilla de política
- Esquema

Modelo de autorización

El modelo de autorización describe el alcance de las solicitudes de autorización realizadas por la aplicación y la base para evaluarlas. Se define en términos de los diferentes tipos de recursos, las acciones que se realizan sobre esos recursos y los tipos de entidad principal que toman esas acciones. También considera el contexto en el que se llevan a cabo esas acciones.

El control de acceso basado en roles (RBAC) es una base de evaluación en la que los roles se definen y asocian a un conjunto de permisos. Después, estas roles se pueden asignar a una o más identidades. La identidad asignada adquiere los permisos asociados al rol. Si se modifican los permisos asociados al rol, la modificación afecta automáticamente a cualquier identidad a la que se haya asignado el rol. Cedar puede respaldar RBAC las decisiones mediante el uso de grupos principales.

El control de acceso basado en atributos (ABAC) es una base de evaluación en la que los permisos asociados a una identidad vienen determinados por los atributos de esa identidad. Cedar puede respaldar ABAC las decisiones mediante el uso de condiciones políticas que hagan referencia a los atributos del director.

El lenguaje de políticas de Cedar permite combinar RBAC y ABAC formar una sola política, ya que permite definir los permisos para un grupo de usuarios, que tienen condiciones basadas en atributos.

Solicitud de autorización

Una solicitud de autorización es una solicitud de Verified Permissions realizada por una aplicación para evaluar un conjunto de políticas a fin de determinar si una entidad principal puede realizar una acción en un recurso en un contexto determinado.

Respuesta de autorización

La respuesta de autorización es la respuesta a la solicitud de autorización. Incluye una decisión de permitir o denegar, además de información adicional, como IDs las políticas determinantes.

Políticas consideradas

Las políticas consideradas son el conjunto completo de políticas que Verified Permissions selecciona para incluirlas al evaluar una solicitud de autorización.

Datos de contexto

Los datos de contexto son valores de atributos que proporcionan información adicional para su evaluación.

Políticas determinantes

Las políticas determinantes son las políticas que determinan la respuesta de autorización. Por ejemplo, si hay dos políticas satisfechas, una de denegación y la otra de permiso, la política de denegación será la política determinante. Si hay varias políticas de permisos satisfechas y ninguna política de prohibición satisfecha, hay varias políticas determinantes. En el caso de que ninguna política coincida y la respuesta sea denegar, no hay políticas determinantes.

Datos de la entidad

Los datos de la entidad son los datos sobre la entidad principal, la acción y el recurso. Los datos de la entidad relevantes para la evaluación de las políticas son la pertenencia a grupos hasta el final de la jerarquía de la entidad y los valores de los atributos de la entidad principal y el recurso.

Permisos, autorizaciones y entidades principales

Verified Permissions administra los permisos y la autorización detallados en las aplicaciones personalizadas que usted crea.

Una entidad principal es el usuario de una aplicación, ya sea humana o automática, que tiene una identidad vinculada a un identificador, como un nombre de usuario o un identificador de máquina. El proceso de autenticación determina si la entidad principal es realmente la identidad que afirma ser.

Asociado a esa identidad hay un conjunto de permisos de aplicación que determinan lo que la entidad principal puede hacer dentro de esa aplicación. La autorización es el proceso de evaluar esos permisos para determinar si una entidad principal está autorizada a realizar una acción determinada en la aplicación. Estos permisos se pueden expresar como políticas.

Aplicación de políticas

La aplicación de las políticas es el proceso de hacer cumplir la decisión de evaluación dentro de la aplicación fuera de Verified Permissions. Si la evaluación de Verified Permissions da como resultado una denegación, la aplicación garantizaría que se impidiera a la entidad principal acceder al recurso.

Almacén de políticas

Un almacén de políticas es un contenedor de políticas y plantillas. Cada almacén contiene un esquema que se utiliza para validar las políticas agregadas al almacén. De forma predeterminada, cada aplicación tiene su propio almacén de políticas, pero varias aplicaciones pueden compartir un único almacén de políticas. Cuando una aplicación realiza una solicitud de autorización, identifica el almacén de políticas utilizado para evaluar esa solicitud. Los almacenes de políticas proporcionan una forma de aislar un conjunto de políticas y, por lo tanto, se pueden usar en una aplicación de varios inquilinos para incluir los esquemas y las políticas de cada inquilino. Una sola aplicación puede tener almacenes de políticas independientes para cada inquilino.

Al evaluar una solicitud de autorización, Verified Permissions solo tiene en cuenta el subconjunto de las políticas del almacén de políticas que son relevantes para la solicitud. La relevancia se determina en función del alcance de la política. El alcance identifica la entidad principal y el recurso específicos a los que se aplica la política, así como las acciones que la entidad principal puede realizar en el recurso. Definir el alcance ayuda a mejorar el rendimiento al reducir el conjunto de políticas consideradas.

Políticas satisfechas

Las políticas satisfechas son las políticas que coinciden con los parámetros de la solicitud de autorización.

Diferencias entre los permisos verificados de Amazon y el lenguaje de la política de Cedar

Amazon Verified Permissions utiliza el motor de lenguaje de políticas de Cedar para realizar las tareas de autorización. Sin embargo, existen algunas diferencias entre la implementación nativa de Cedar y la implementación de Cedar en Verified Permissions. En este tema se explican esas diferencias.

Definición de espacio de nombres

La implementación de Verified Permissions de Cedar presenta las siguientes diferencias con respecto a la implementación nativa de Cedar:

- Verified Permissions solo admite un espacio de nombres en un esquema definido en un almacén de políticas.

- Verified Permissions no permite crear un espacio de nombres con los siguientes valores: `aws`, `amazon` o `cedar`.

Compatibilidad con las plantillas de política

Tanto Verified Permissions como Cedar solo permiten marcadores de posición en el ámbito para `principal` y `resource`. Sin embargo, Verified Permissions también requiere que ni `principal` ni `resource` carezcan de restricciones.

La siguiente política es válida en Cedar, pero Verified Permissions la rechaza porque la `principal` no tiene restricciones.

```
permit(principal, action == Action::"view", resource == ?resource);
```

Los dos ejemplos siguientes son válidos tanto en Cedar como en Verified Permissions porque la `principal` y el `resource` tienen restricciones.

```
permit(principal == User::"alice", action == Action::"view", resource == ?resource);
```

```
permit(principal == ?principal, action == Action::"a", resource in ?resource);
```

Compatibilidad con esquemas

Los permisos verificados requieren que todos los nombres de las JSON claves del esquema no sean cadenas vacías. Cedar permite cadenas vacías en algunos casos, como en el caso de las propiedades.

Compatibilidad con tipos de extensión

Verified Permissions admite los tipos de extensión de Cedar en las políticas, pero actualmente no permite incluirlos en la definición de un esquema o como parte del parámetro `entities` de las operaciones `IsAuthorized` y `IsAuthorizedWithToken`.

Los tipos de extensión incluyen los tipos de datos de punto fijo (`decimal`) y dirección IP (`ipaddr`).

JSONFormato Cedar para entidades

En este momento, los permisos verificados requieren que pase la lista de entidades que se van a tener en cuenta en una solicitud de autorización utilizando la estructura definida para ellos

EntitiesDefinition, que es una matriz de EntityItemelementos. Actualmente, Verified Permissions no permite aprobar la lista de entidades que deben tenerse en cuenta en una solicitud de autorización en JSONformato Cedar. Para conocer los requisitos específicos sobre el formato de las entidades para su uso en Verified Permissions, consulte Formato de entidades en las políticas de Amazon Verified Permissions.

Definición de grupos de acción

Los métodos de autorización de Cedar requieren una lista de las entidades que se deben tener en cuenta al evaluar una solicitud de autorización con respecto a las políticas.

Puede definir las acciones y los grupos de acciones que utiliza su aplicación en el esquema. Sin embargo, Cedar no incluye el esquema como parte de una solicitud de evaluación. En su lugar, Cedar usa el esquema solo para validar las políticas y las plantillas de políticas que envíe. Dado que Cedar no hace referencia al esquema durante las solicitudes de evaluación, incluso si ha definido grupos de acción en el esquema, también debe incluir la lista de todos los grupos de acciones como parte de la lista de entidades que debe pasar a las API operaciones de autorización.

Verified Permissions lo hace por usted. Todos los grupos de acciones que defina en su esquema se anexan automáticamente a la lista de entidades que pase como parámetro de las operaciones `IsAuthorized` o `IsAuthorizedWithToken`.

Límites de longitud y tamaño

Verified Permissions admite el almacenamiento en forma de almacenes de políticas para almacenar esquemas, políticas y plantillas de políticas. Ese almacenamiento hace que Verified Permissions imponga algunos límites de longitud y tamaño que no son relevantes para Cedar.

Objeto	Límite de Verified Permissions (en bytes)	Límite de Cedar
Tamaño de la política[1]	10 000	Ninguna
Descripción de la política insertada	150	No aplicable a Cedar
Tamaño de la plantilla de política	10 000	Ninguna

Objeto	Límite de Verified Permissions (en bytes)	Límite de Cedar
Tamaño del esquema	10 000	Ninguna
Tipo de identidad	200	Ninguna
ID de política	64	Ninguna
ID de plantilla de política	64	Ninguna
ID de la identidad	200	Ninguna
ID del almacén de políticas	64	No aplicable a Cedar

[1] Existe un límite de políticas por almacén de políticas en Verified Permissions en función del tamaño combinado de las entidades principales, las acciones y los recursos de las políticas creadas en el almacén de políticas. El tamaño total de todas las políticas pertenecientes a un único recurso no puede superar los 200 000 bytes. En el caso de las políticas vinculadas a plantillas, el tamaño de la plantilla de política se cuenta solo una vez, más el tamaño de cada conjunto de parámetros utilizado para crear una instancia de cada política vinculada a una plantilla.

Historial de documentos de la Guía del usuario de Amazon Verified Permissions

En la siguiente tabla se describen las versiones de la documentación de Verified Permissions.

Cambio	Descripción	Fecha
Fuentes de identidad del OIDC	Ahora puede autorizar a los usuarios de los proveedor es de identidad de OpenID Connect (OIDC).	8 de junio de 2024
Autorización por lotes con tokens de origen de identidad	Ahora puede autorizar a los usuarios de un grupo de usuarios de Amazon Cognito en una sola solicitud de `BatchIsAuthorizedW ithToken` API.	5 de abril de 2024
Creación de un almacén de políticas con API Gateway	Ahora puede crear un almacén de políticas a partir de una API existente y un grupo de usuarios de Amazon Cognito.	1 de abril de 2024
Conceptos y ejemplos de contexto	Se agregó información sobre el contexto en las solicitudes de autorización con permisos verificados.	1 de febrero de 2024
Conceptos y ejemplos de autorización	Se agregó información sobre las solicitudes de autorización con permisos verificados.	1 de febrero de 2024
AWS CloudFormation integrati on	Verified Permissions permite crear fuentes de identidad , políticas, almacenes de políticas y plantillas de	30 de junio de 2023

	políticas en AWS CloudFormation.	
Versión inicial	Versión inicial de la Guía del usuario de Amazon Verified Permissions	13 de junio de 2023

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.