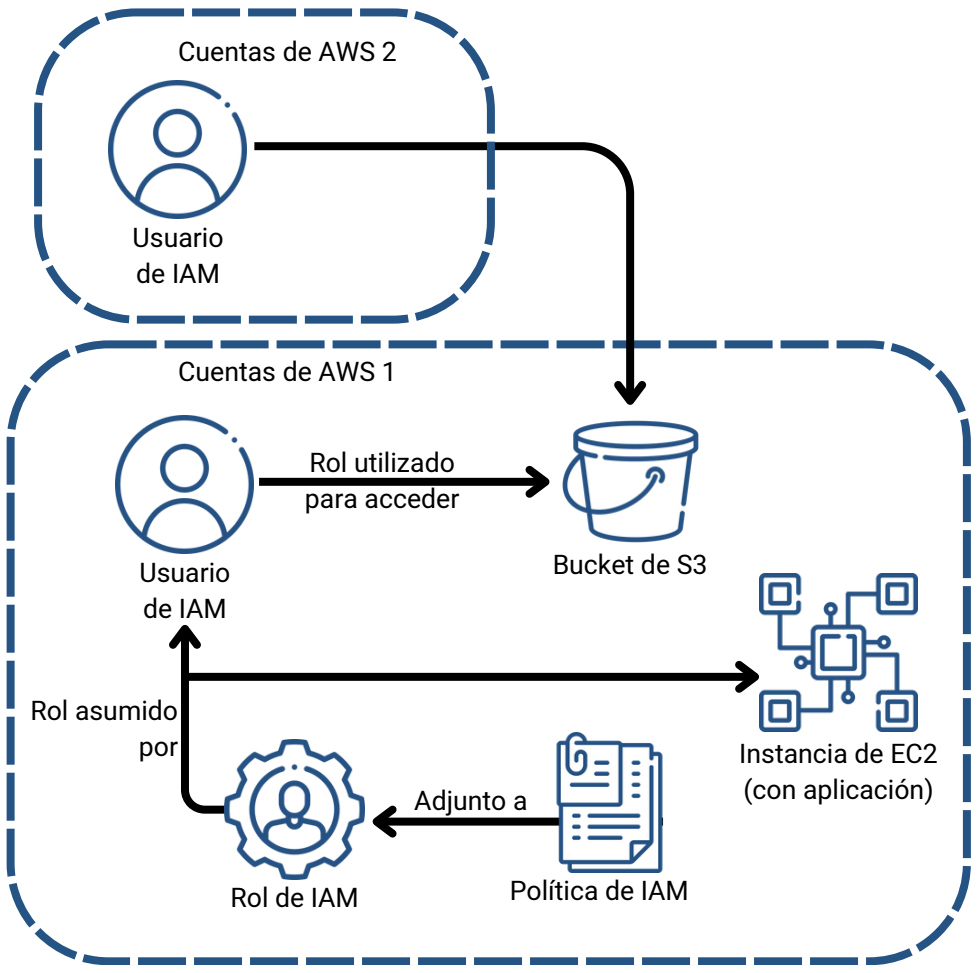
Lección 4

Federación de usuarios

Roles de IAM



Características del rol de IAM

- Proporciona credenciales de seguridad temporales.
- No se encuentra asociado únicamente con una persona
- Puede ser asumido por una persona, aplicación o servicio
- A menudo se utiliza para delegar el acceso

Casos prácticos

- Proporcionar recursos de AWS con acceso a servicios de AWS
- Proporcionar acceso a usuarios autenticados externamente
- Proporcionar acceso a terceros
- Cambiar de roles para acceder a recursos en -
  - Su cuenta de AWS
  - Cualquier otra cuenta de AWS (acceso entre cuentas)

Un rol de IAM le permite definir un conjunto de permisos para acceder a los recursos que un usuario o servicio necesita. Sin embargo, los permisos no están adjuntos a ningún usuario o grupo de IAM. En cambio, los permisos se adjuntan a un rol, y el rol lo asume el usuario o servicio.

Cuando un usuario asume un rol, sus permisos anteriores se olvidan temporalmente. AWS devuelve las credenciales de seguridad temporales que el usuario o la aplicación pueden luego utilizar para realizar solicitudes programáticas a AWS.

Al utilizar roles de IAM, no tiene que compartir las credenciales de seguridad a largo plazo para cada entidad que requiere acceso a un recurso como, por ejemplo, crear un usuario de IAM.

En el caso de un servicio como Amazon EC2, las aplicaciones o los servicios de AWS pueden asumir un rol de manera programática en tiempo de ejecución.

La entidad principal que asume el rol podría ser un usuario, grupo o rol de IAM de otra cuenta de AWS, incluidas las cuentas que no son de su propiedad.

Al crear un rol para el acceso a cuentas externas, no necesita administrar nombres de usuario y contraseñas para terceros. Si ya no desea que alguien o algún sistema tenga acceso, puede modificar o eliminar el rol. Por lo tanto, no es necesario crear ni administrar cuentas para personas ajenas a su organización.

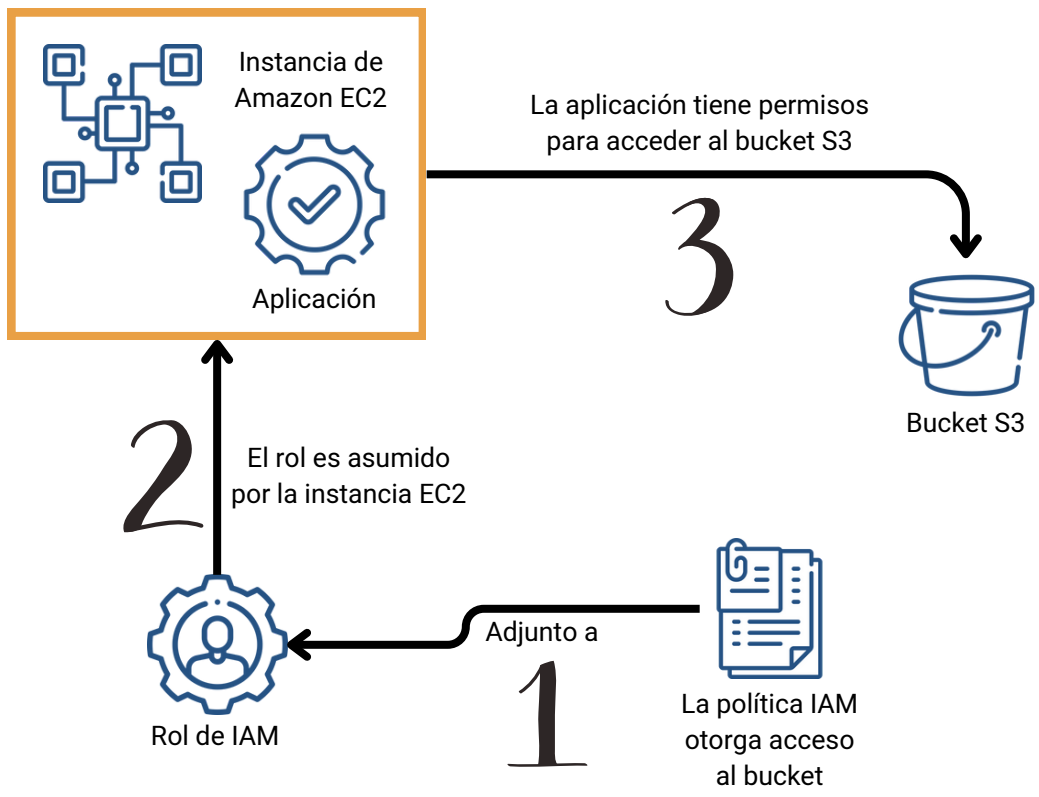
Resumen: Demostración de perfil de instancia EC2

Escenario:

- Una aplicación que se ejecuta en una instancia EC2 necesita acceso a un bucket S3

Solución:

- Definir una política IAM que otorgue acceso al bucket S3.
- Adjuntar la política a un rol.
- Permitir que la instancia EC2 asuma el rol.



Este diagrama ilustra la demostración dirigida por el educador.

- Una aplicación se ejecuta en una instancia EC2 y esa aplicación necesita acceder al bucket S3.
- Un administrador crea un rol IAM.
- Luego, crean una política IAM que otorga acceso de solo lectura al bucket S3 especificado. La política también incluye una política de confianza que permite a la instancia EC2 asumir el rol y obtener las credenciales temporales.
- A continuación, adjuntan la política IAM al rol.

Cuando la aplicación se ejecuta en la instancia, puede asumir el rol y usar las credenciales temporales del rol para acceder al bucket.

Con esta arquitectura, el administrador no necesita otorgar permiso directamente al desarrollador de la aplicación para acceder al bucket, y el desarrollador nunca necesita compartir ni administrar credenciales.

Otorgar permisos para asumir un rol

Para que un usuario, aplicación o servicio de IAM asuma un rol, debe otorgar permisos para cambiar al rol AWS Security Token Service (AWS STS)

- Servicio web que permite solicitar credenciales temporales con privilegios limitados
- Las credenciales pueden ser utilizadas por los usuarios de IAM o por usuarios que usted autentique (usuarios federados)

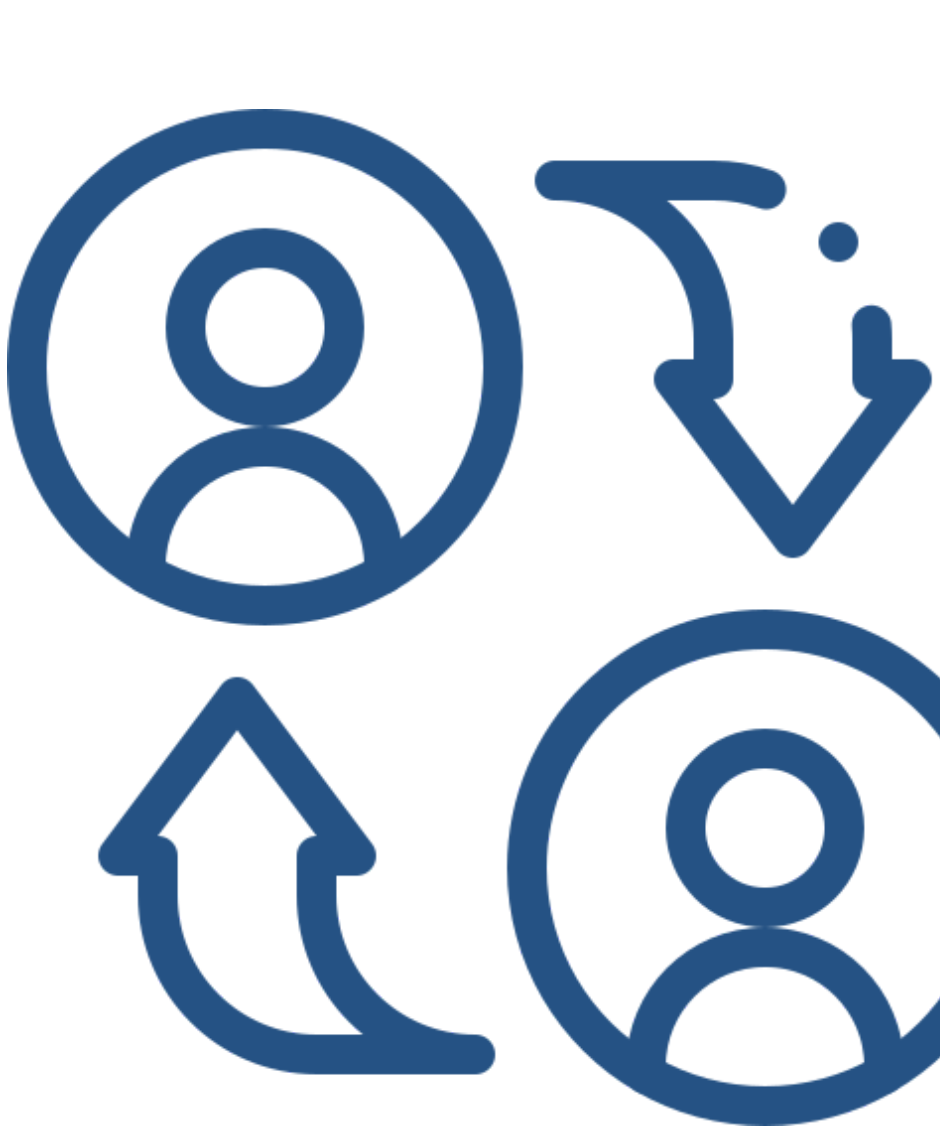
Política de ejemplo: permite que un usuario de IAM asuma un rol

```
{
   "Version": "2012-10-17",
   "Statement": {
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource":  "arn:aws:iam::123456789012:role/Test*"
   }
}
```

AWS Security Token Service (AWS STS)

AWS Security Token Service también se conoce como AWS STS. Esun servicio web que permite a un usuario de IAM, un usuario federado o una aplicación asumir el rol de IAM que desee.

Cuando la operación AssumeRole de la API de AWS STS se invoca correctamente, el servicio web devuelve las credenciales temporales con privilegios limitados que solicitó el usuario de IAM o el usuario que se autenticó a través de federación. Por lo general, la operación AssumeRole se utiliza para acceso entre cuentas o para federación.

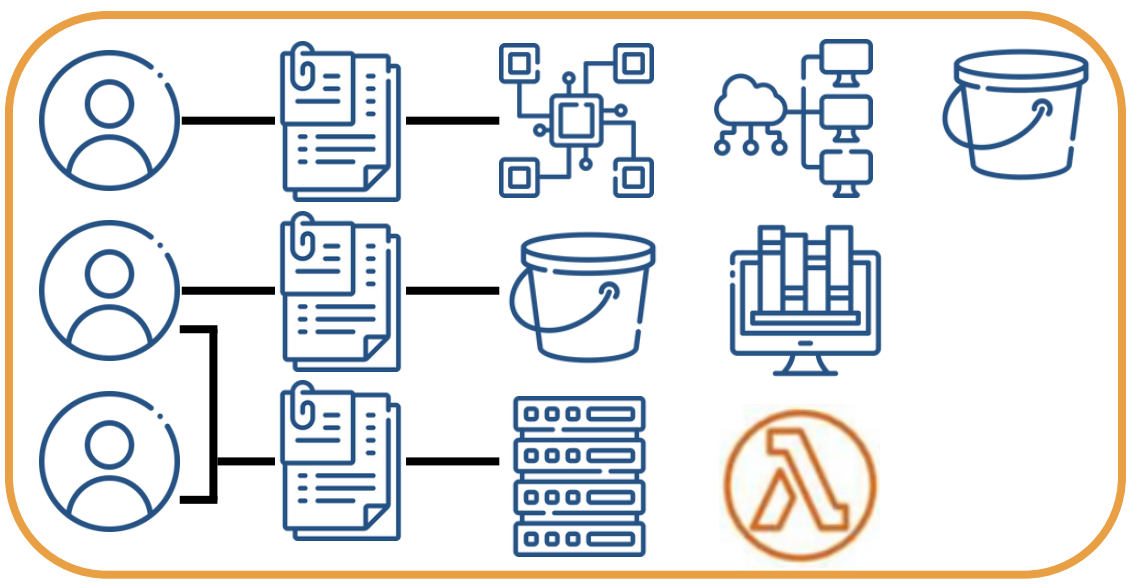


La política de ejemplo permite a un usuario de IAM asumir cualquier rol definido en el número de cuenta de AWS 123456789012, siempre que el nombre del rol comience con Test.

Control de acceso basado en roles (RBAC)

Enfoque tradicional para control de acceso:

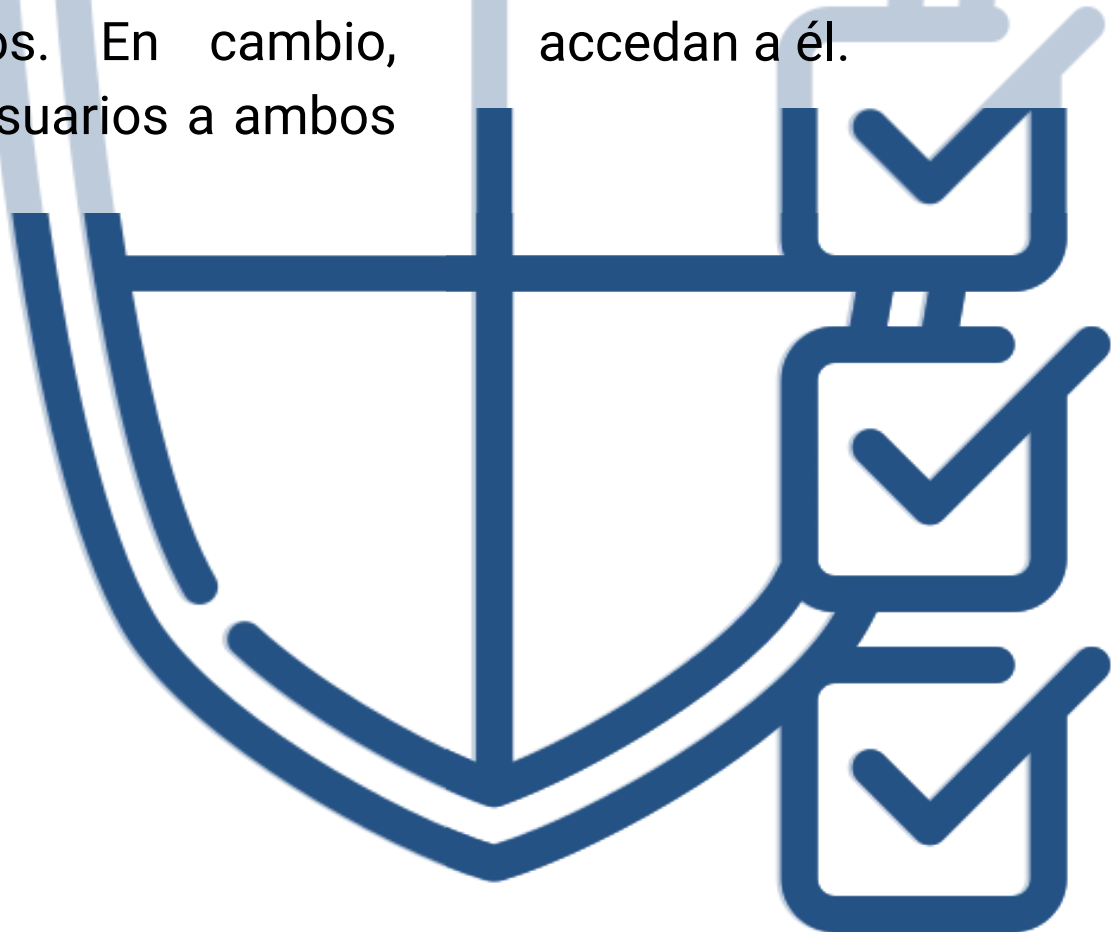
- Otorgar a los usuarios permisos específicos basados en la función del trabajo (como, por ejemplo, administrador de base de datos)
- Crear un rol de IAM distinto para cada combinación de permisos
- Actualizar permisos agregando acceso para cada nuevo recursos (puede llevar mucho tiempo seguir actualizando las políticas)



Ahora considerará dos enfoques diferentes para el control de acceso: control de acceso basado en roles (RBAC) y control de acceso basado en atributos (ABAC). Primero aprenderá sobre RBAC.

Históricamente, RBAC se ha utilizado en las instalaciones y en la nube. Con este modelo, otorga a los usuarios acceso explícito a un conjunto de permisos. Supongamos que tiene administradores de bases de datos, administradores de red y desarrolladores. Si tiene uno o más administradores de red que también sean desarrolladores, no creará una nueva política para otorgar esos permisos. En cambio, agrega esos usuarios a ambos roles.

Este enfoque es familiar y tiene muchas ventajas. Sin embargo, la persona que mantiene los permisos en este modelo puede encontrarse con que debe actualizar constantemente los archivos de permisos para agregar acceso a determinados roles cada vez que se crea un nuevo recurso. Por ejemplo, deben actualizar una política con un ARN cada vez que alguien crea un nuevo recurso y quiere permitir que los usuarios accedan a él.



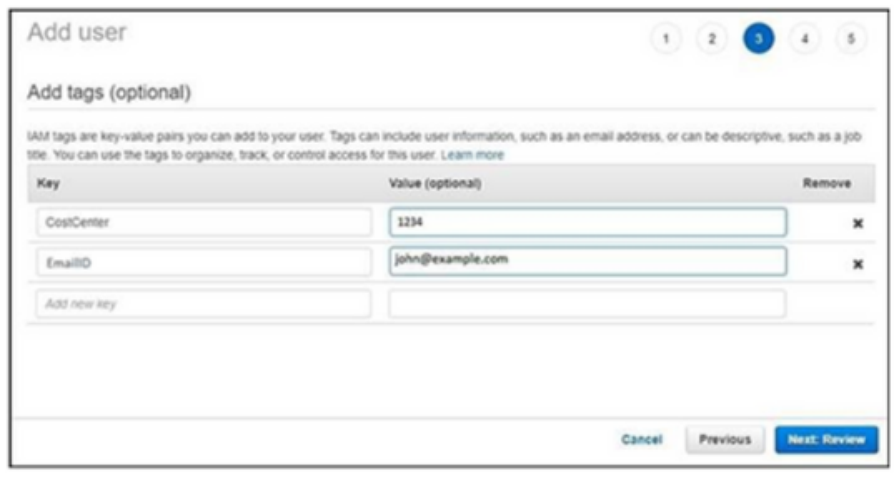
Práctica recomendada:etiquetado

Una etiqueta consta de un nombre y (opcionalmente) un valor
- Se puede aplicar a recursos en todas sus cuentas de AWS
- Las claves y los valores de las etiquetas se entregan a través de diferentes operaciones API

Definir etiquetas personalizadas
Múltiples usos prácticos
- Facturación, vistas filtradas, control de acceso, etc.

Etiquetas de ejemplo aplicadas a una instancia EC2:
- Nombre = servidor web
- Proyecto = univornio
- Pila = dev

Las etiquetas también se pueden aplicar a ususarios de IAM o roles de IAM, por ejemplo



Antes de considerar el segundo enfoque para los controles de permisos, debe comprender la función de etiquetado en AWS.

Amazon permite a los clientes asignar metadatos a sus recursos e identidades de AWS en forma de etiquetas. Cada etiqueta es una etiqueta simple que consta de una clave definida por el cliente y un valor opcional. Las etiquetas pueden facilitar el proceso de administrar, buscar y filtrar los recursos.

Las etiquetas tienen muchos usos prácticos. Por ejemplo, puede crear etiquetas técnicas para identificar que un recurso es un servidor web, parte de un proyecto específico, parte de un entorno específico (prueba, desarrollo o producción), entre otros. También puede crear etiquetas empresariales para identificar el departamento o centro de costos que se debe facturar por este recurso o el proyecto del que forma parte este recurso. Por último, también puede configurar etiquetas de seguridad, como un identificador para el nivel de confidencialidad de datos específico que admite un recurso.

Puede crear hasta etiquetas por recurso. Para cada recurso, cada clave de etiqueta debe ser única y cada etiqueta solo puede tener un valor. Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.

También puede agregar etiquetas a usuarios de IAM y roles de IAM. Las etiquetas son una parte importante del segundo método de control de acceso que aprenderá a continuación.

Control de acceso basado en atributos (ABAC)

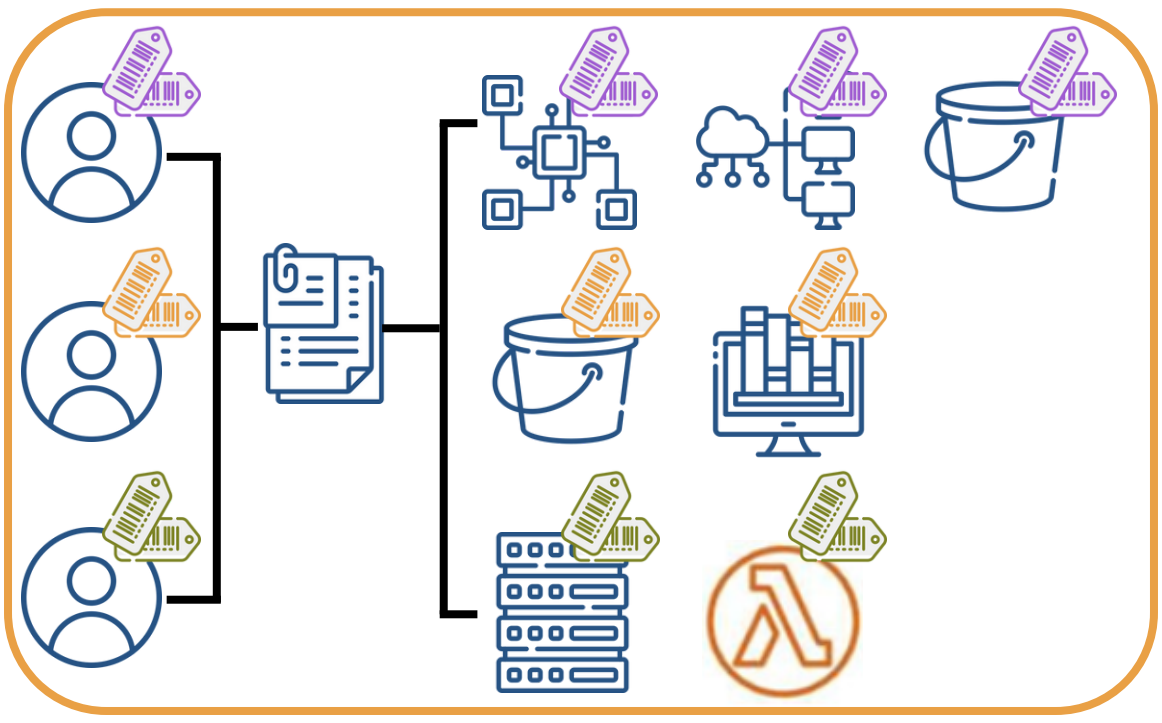
Enfoque altamente escalable para control de acceso

- Los atributos son una clave o un par clave-valor, como por ejemplo una etiqueta
- Ejemplo de atributos -
  - Equipo =Desarrolladores
  - Proyecto = Unicornio

Las reglas de permisos (política) son más fáciles de mantener con ABAC que con RBAC

Beneficios

- Los permisos se aplican automáticamente, según los atributos
- Los permisos granulares son posibles sin una actualización de permisos para cada usuario o recurso nuevo
- Completamente auditable



Ahora que conoce la función de etiquetado, conocerá el segundo enfoque de control de acceso: el control de acceso basado en atributos (ABAC).

ABAC le permite utilizar atributos para crear reglas de permisos generales que escalan con su organización.

En este modelo, los usuarios de IAM tienen atributos que usted creó y aplicó, como una o más etiquetas.

Los recursos también tienen atributos, como etiquetas coincidentes, que usted también aplicó a los recursos.

Con el enfoque RBAC, los permisos de escritura son relativamente sencillos. La política comprueba si un atributo que se aplica al usuario de IAM también se aplica al recurso al que desea acceder. Cuando crea nuevos usuarios de IAM y nuevos recursos de cuenta, aplica las etiquetas correctas a los usuarios y a los recursos.

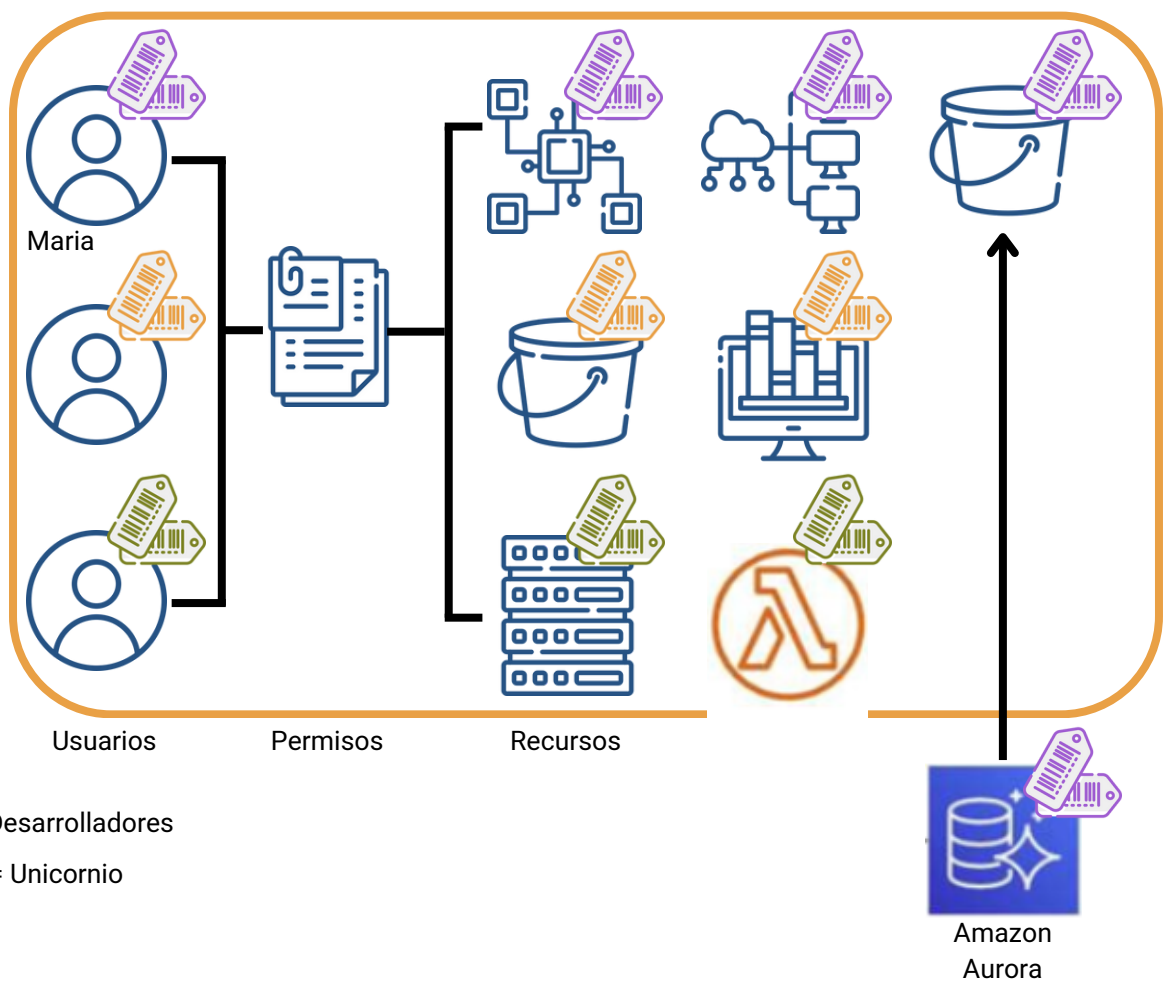
Con el enfoque ABAC, puede otorgar a los desarrolladores acceso a los recursos de su proyecto, pero no es necesario especificar recursos en el archivo de la política.

Puede imaginarse lo escalable que puede resultar el enfoque ABAC para la administración de acceso. No es necesario modificar la configuración de permisos. Los permisos se aplican automáticamente cuando se crean recursos o usuarios con las etiquetas correctas.

Aplicar ABAC a su organización

Cómo aplicar ABAC a su organización

1. Establecer atributos de control de acceso en identidades
2. Requerir atributos para recursos nuevos
3. Configurar permisos basados en atributos
4. Probar
   a. Crear recursos nuevos
   b. Verificar que los permisos se aplican automáticamente



Equipo = Desarrolladores

Proyecto = Unicornio

Para aplicar ABAC a su organización, el primer paso es crear identidades, como usuarios de IAM o roles de IAM. Estas identidades deben tener los atributos que se utilizarán para fines de control de acceso. Por ejemplo, puede aplicar las etiquetas Equipo = Desarrolladores y Proyecto = Unicornio al usuario Maria.

Luego, solicite atributos para recursos nuevos. Debe crear políticas que hagan cumplir las reglas. Por ejemplo, podría requerir que un atributo Proyecto y un atributo Equipo se apliquen a cualquier recurso cuando se cree.

En tercer lugar, configure permisos de acceso según los atributos. Por ejemplo, supongamos que un usuario de IAM tiene las etiquetas Proyecto = Unicornio y Equipo = Desarrolladores. Si ese usuario intenta acceder a un recurso que tiene valores coincidentes para las mismas dos etiquetas, entonces la política permitirá el acceso. De lo contrario, la política denegará el acceso.

Cuarto, pruebe su configuración. Por ejemplo, podría intentar crear una instancia de base de datos de Amazon Aurora sin las etiquetas requeridas. El intento debería fallar. Intente crear la instancia de la base de datos nuevamente con las etiquetas requeridas. Esta vez, debería poder crear el recurso correctamente. Finalmente, podría intentar acceder a la instancia de la base de datos como usuario María. Debería poder acceder correctamente. Sin embargo, se le debe denegar el acceso si intenta acceder a la instancia de la base de datos como un usuario diferente que no tiene las etiquetas coincidentes.

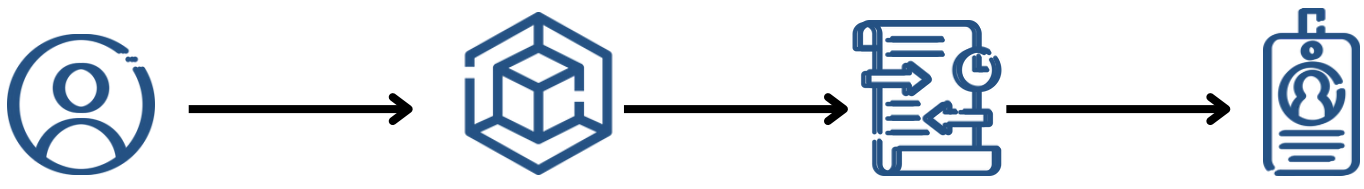
Usuarios autenticados externamente

Federación de identidades

- Autenticación del usuario completada por un sistema que es externo a la cuenta de AWS
  - Ejemplo: directorio corporativo
- Proporciona una forma de permitir el acceso a través de las identidades existentes, sin crear usuarios de IAM

Opciones de federación de identidades

1. AWS STS
   - Proveedores de servicio de identidades públicas (IdP)
   - Aplicación de agente de identidades personalizada
2. Lenguaje de marcado para confirmaciones de seguridad (SAML)
3. Amazon Cognito



El usuario accede al agente de identidades a través de la aplicación

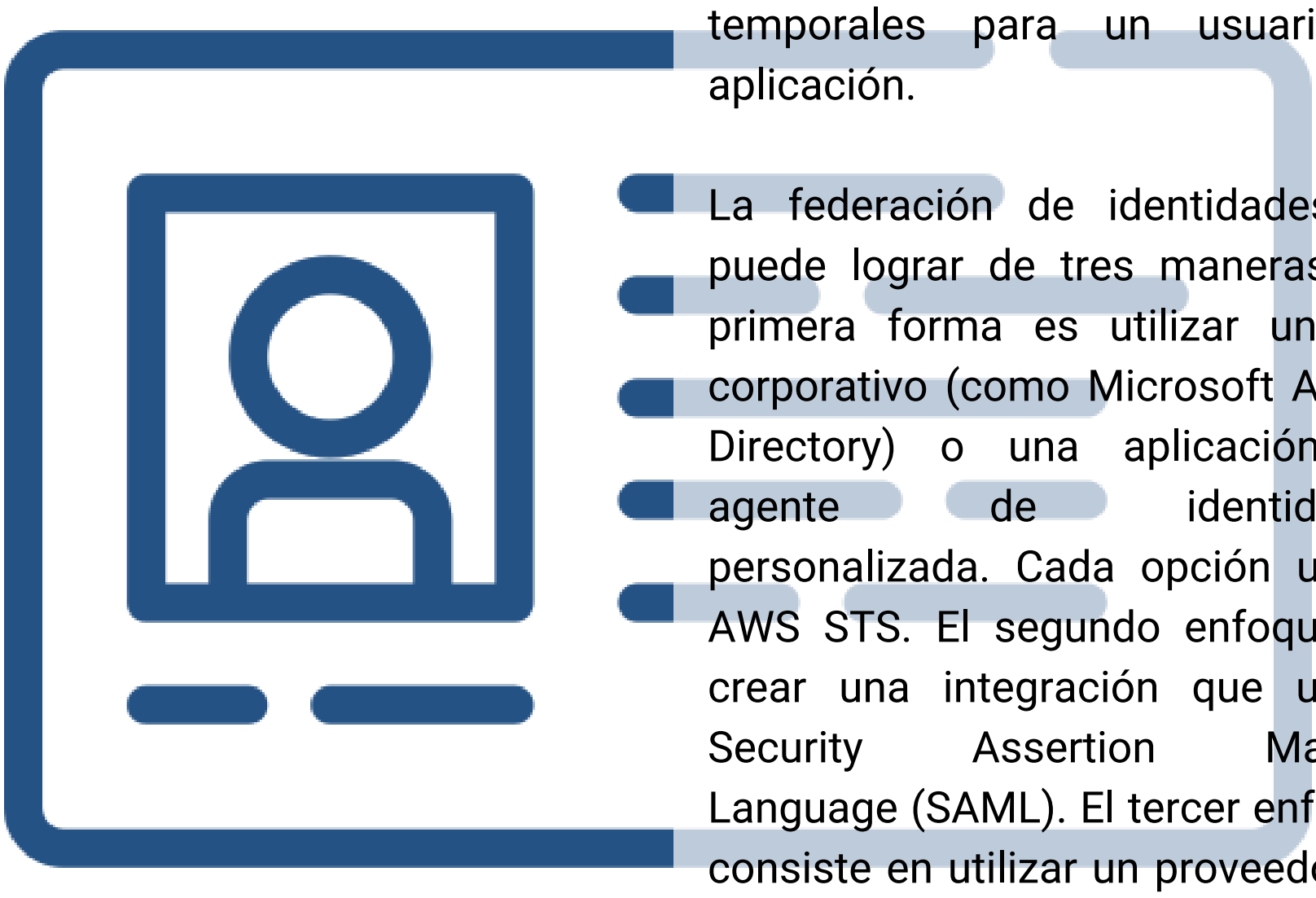
El agente de identidades autentica al usuario

Solicita credenciales temporales de AWS STS

Credenciales temporales devueltas a la aplicación

Ahora aprenderá sobre un tema nuevo: usuarios autenticados externamente.

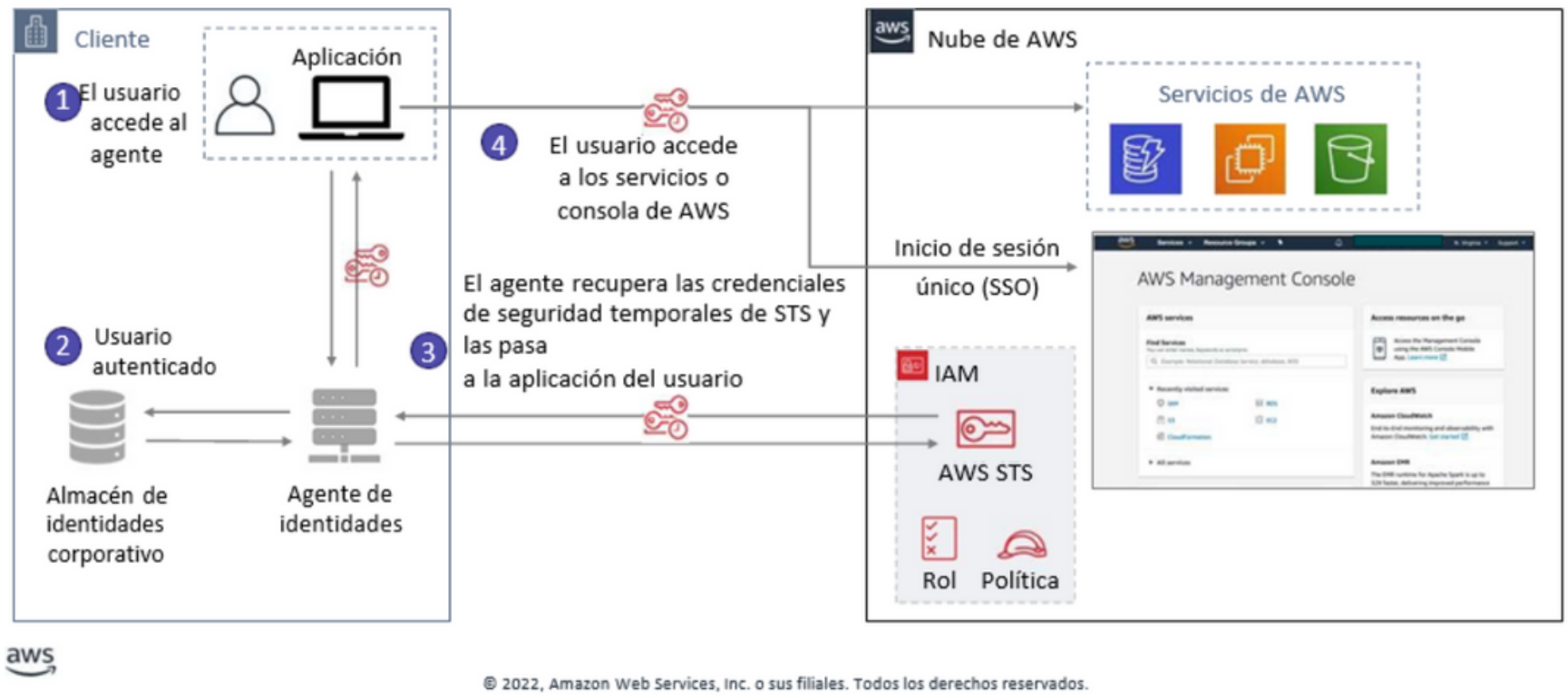
IAM admite la federación de identidades para el acceso delegado a la Consola de administración de AWS o a las API de AWS. Con la federación de identidades, se otorga a las identidades externas acceso seguro a los recursos de su cuenta de AWS sin necesidad de crear usuarios de IAM.



El gráfico muestra los cuatro pasos principales que ocurren cuando utiliza un proveedor de identidad (IdP) para crear credenciales temporales para un usuario o aplicación.

La federación de identidades se puede lograr de tres maneras. La primera forma es utilizar un IdP corporativo (como Microsoft Active Directory) o una aplicación de agente de identidades personalizada. Cada opción utiliza AWS STS. El segundo enfoque es crear una integración que utilice Security Assertion Markup Language (SAML). El tercer enfoque consiste en utilizar un proveedor de identidades web, como Amazon Cognito. Las siguientes diapositivas analizan cada uno de estos tres enfoques.

Federación de identidades con un agente de identidades



Ahora aprenderá cómo lograr la federación de identidades mediante el uso de un agente de identidades.

El proceso incluye estos pasos:

1. Un usuario accede a una aplicación. El usuario ingresa su ID de usuario y contraseña y los envía
2. El agente de identidades recibe la solicitud de autenticación. Luego, se comunica con el almacén de identidades corporativas, que puede ser Microsoft Active Directory o un servidor Lightweight Directory Access Protocol (LDAP).
3. Si la solicitud de autenticación se realizó correctamente, el agente de identidades realiza una solicitud a AWS STS. La solicitud es para recuperar credenciales de seguridad temporales de AWS para la aplicación del usuario.
4. La aplicación de usuario recibe las credenciales de seguridad temporales de AWS y redirige al usuario a la Consola de administración de AWS. El usuario no necesitaba iniciar sesión directamente en AWS con un conjunto diferente de credenciales. Este proceso es un ejemplo de implementación de inicio de sesión único (SSO). La aplicación de usuario también podría utilizar estas mismas credenciales de seguridad temporales de AWS para acceder a los servicios de AWS si el documento de política de IAM lo permite.

Federación de identidades a través de SAML

Ahora conocerá la segunda opción para lograr la federación de identidades. Este enfoque utiliza el estándar abierto SAML para intercambiar datos de autenticación y autorización entre IdP y proveedores de servicios.

El proceso incluye estos pasos:

1. Un usuario de su organización navega a un portal interno de su red. El portal también actúa como IdP que administra la confianza SAML entre su organización y AWS.
2. El IdP autentica la identidad del usuario en el almacén de identidades, que puede ser un servidor LDAP o Microsoft Active Directory.
3. El portal recibe la respuesta de autenticación como una confirmación SAML del IdP.
4. El cliente publica la confirmación SAML en el punto de enlace de inicio de sesión de AWS para SAML. El punto de enlace se comunica con AWS STS e invoca la operación AssumeRoleWithSAML para solicitar credenciales de seguridad temporales y crear una URL de inicio de sesión.
5. El cliente recibe las credenciales de seguridad temporales de AWS. El cliente es redirigido a la Consola de administración de AWS y se autentica con las credenciales de seguridad temporales de AWS.

Amazon Cognito

Amazon Cognito es un servicio completamente administrado

- Proporciona autenticación, autorización y administración de usuarios para sus aplicaciones web y móviles
- Amazon Cognito proporciona federación de identidades web
  - Se pueden utilizar como el agente de identidades que admite IdP que sean compatibles con OpenID Connect (OIDC)

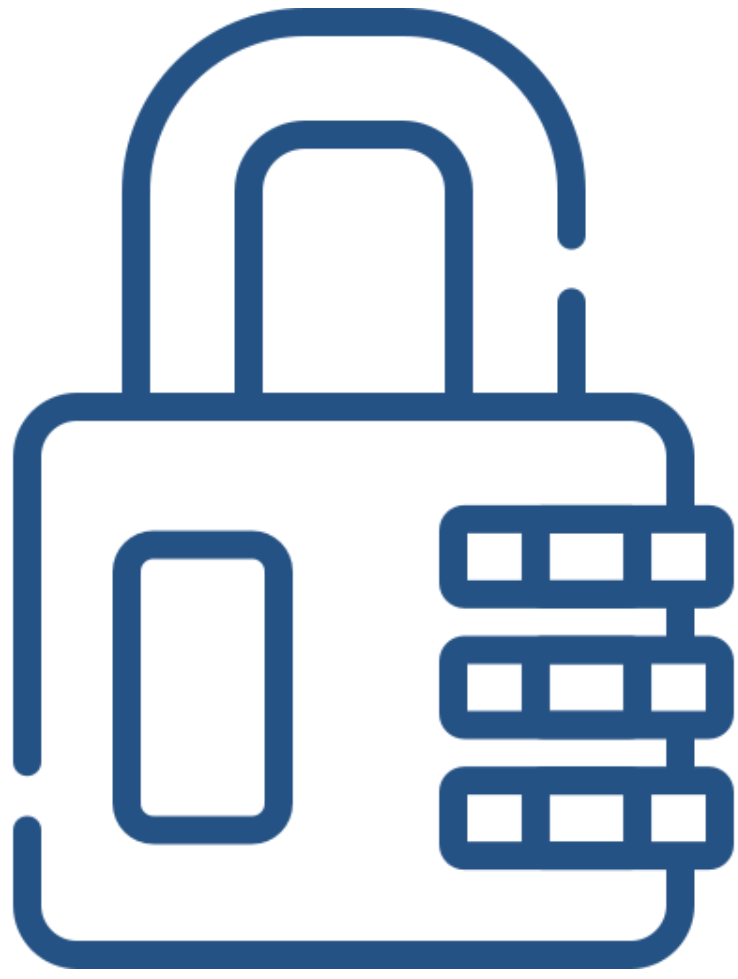
Identidades federadas

- Los usuarios inician sesión con proveedores de identidad social (Amazon, Facebook, Google) o con SAML

Grupos de usuarios

- Puede mantener un directorio con tokens de actualización de perfiles de usuario

La tercera y última opción de federación de identidades es utilizar Amazon Cognito. Amazon Cognito es un servicio totalmente administrado que proporciona autenticación, autorización y administración de usuarios para aplicaciones web y móviles. Los usuarios pueden iniciar sesión directamente con un nombre de usuario y contraseña o mediante un tercero, como Facebook, Amazon o Google.

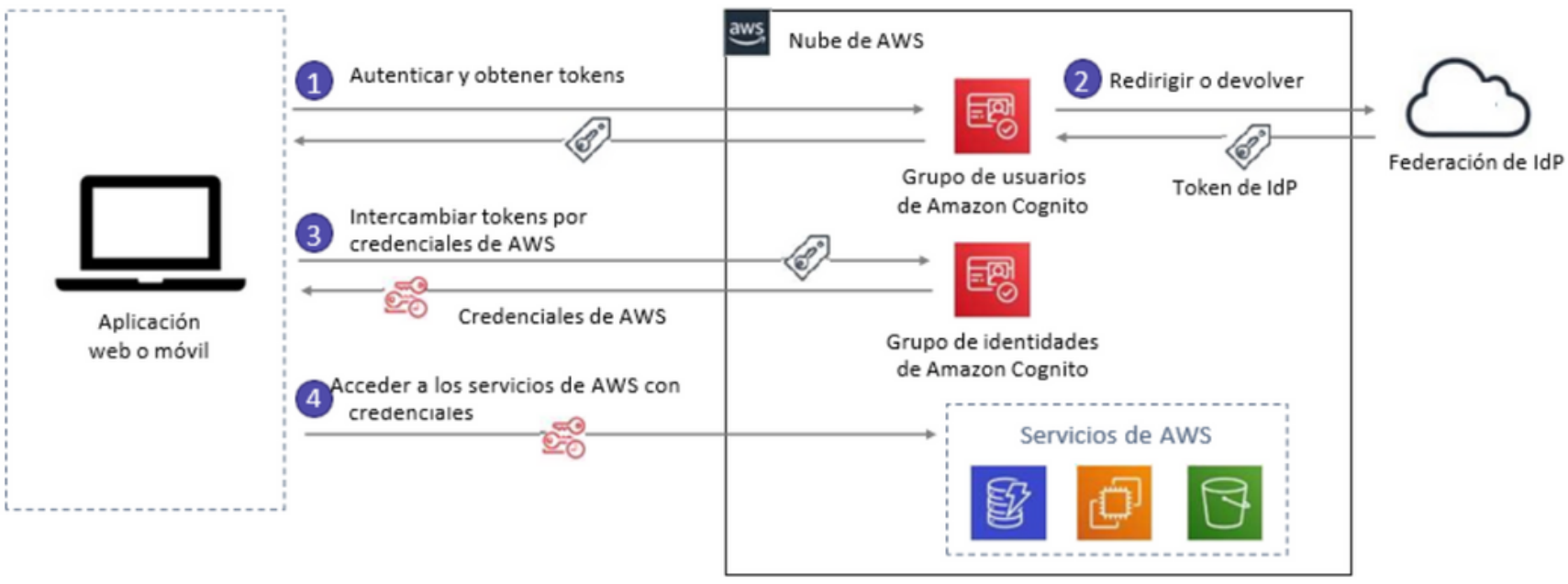


Los dos componentes principales de Amazon Cognito son los grupos de usuarios y los grupos de identidades.

Un grupo de usuarios es un directorio de usuarios de Amazon Cognito. Con un grupo de usuarios, los usuarios pueden iniciar sesión en su aplicación web o móvil por medio de Amazon Cognito.
También pueden federarse a través de un IdP de terceros. Todos los miembros del grupo de usuarios tienen un perfil en el directorio al que se puede acceder mediante un SDK.

Los grupos de identidades permiten la creación de identidades únicas y la asignación de permisos para los usuarios. Con un grupo de identidades, los usuarios pueden obtener credenciales temporales de AWS para acceder a los servicios o recursos de AWS. Los grupos de identidades pueden comunicarse con el inicio de sesión social de los grupos de usuarios de Amazon Cognito con Facebook, Google, e iniciar sesión con Amazon; y proveedores de OpenID Connect (OIDC).

Ejemplo de Amazon Cognito



En este escenario, el objetivo es autenticar a un usuario a través de Amazon Cognito y luego otorgarle a ese usuario acceso a otro servicio de AWS.

- En el primer paso, el usuario de la aplicación inicia sesión mediante un grupo de usuarios de Amazon Cognito y luego de autenticarse exitosamente recibe tokens del grupo de usuarios.
- Luego, la aplicación intercambia los tokens del grupo de usuarios por credenciales de AWS mediante un grupo de identidades.
- Por último, el usuario de la aplicación utiliza esas credenciales de AWS para acceder a otros servicios de AWS.

Entre los aprendizajes clave de esta lección de esta unidad, se incluyen los siguientes:

- Los roles de IAM proporcionan credenciales de seguridad temporales que puede asumir una persona, aplicación o servicio.
- El AWS Security Token Service (STS) le permite solicitar credenciales temporales de AWS.
- Con la federación de identidades, la autenticación del usuario se produce en forma externa a la cuenta de AWS.
    - Se logra a través de STS, SAML o Amazon Cognito.