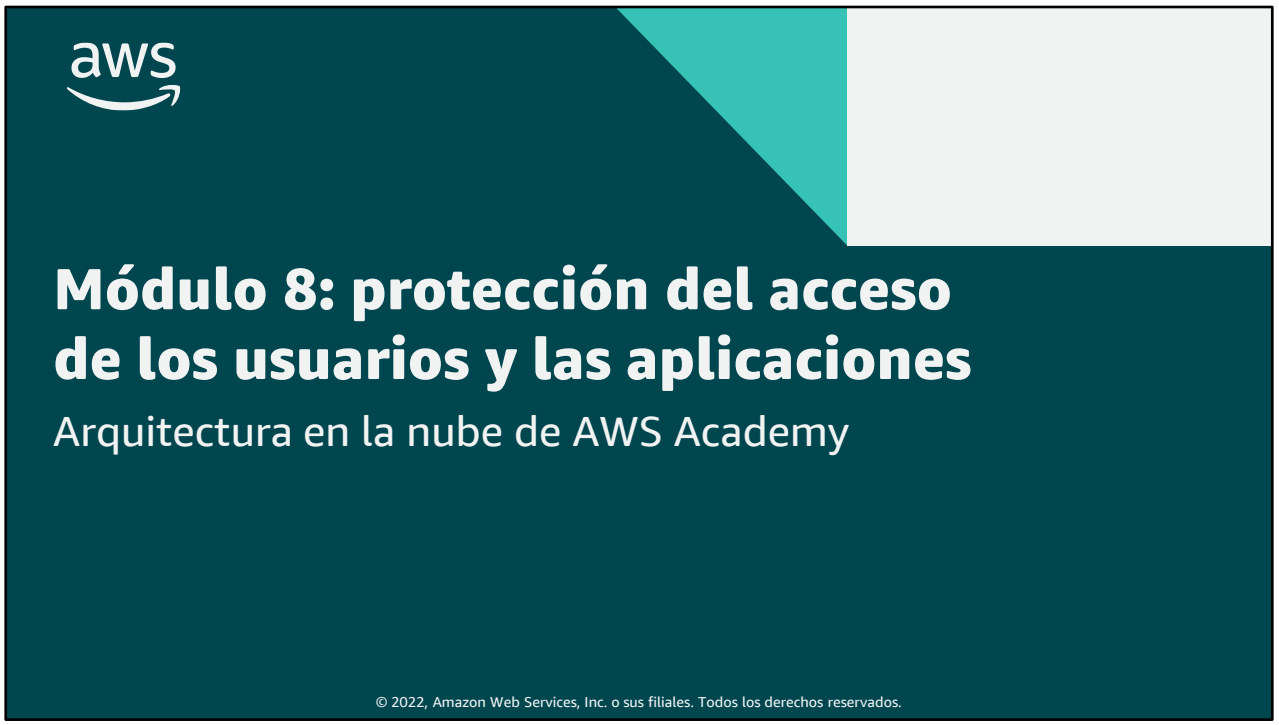
AWS Academy Cloud Architecting (LA)

Module 08 Student Guide

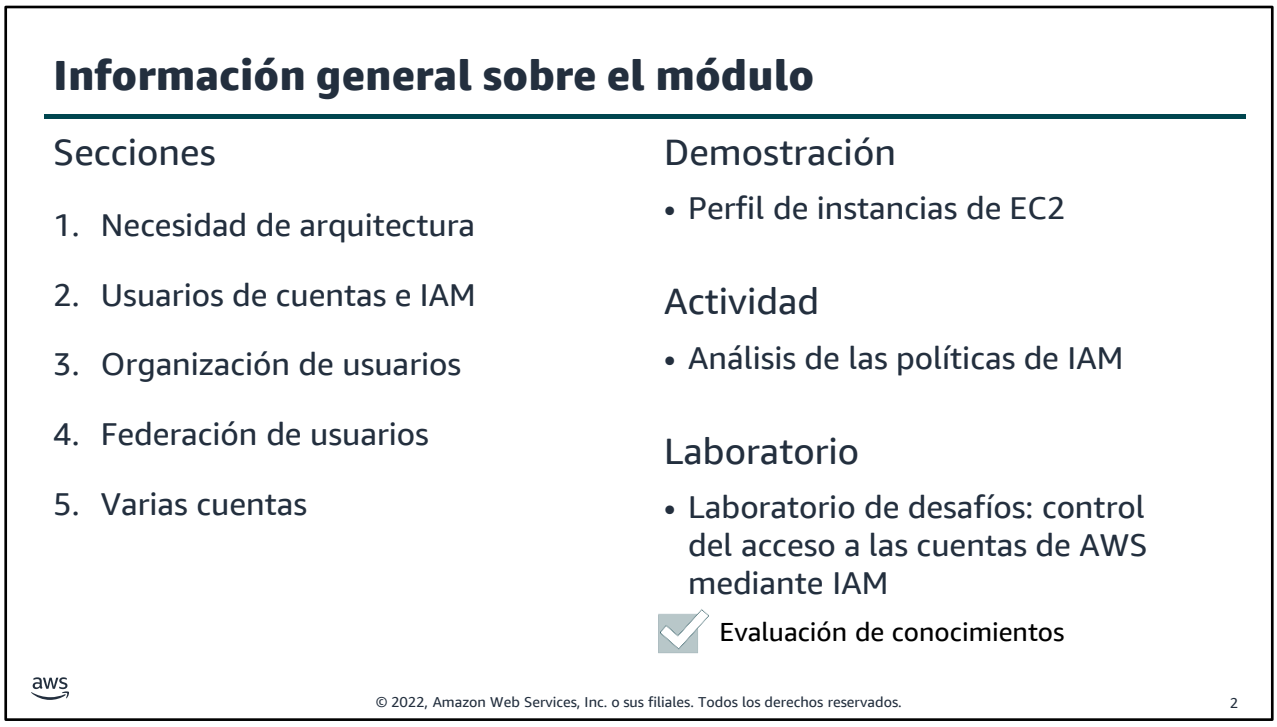
Versión 2.0.13

200-ACACAD-20-LA-SG

Contenido



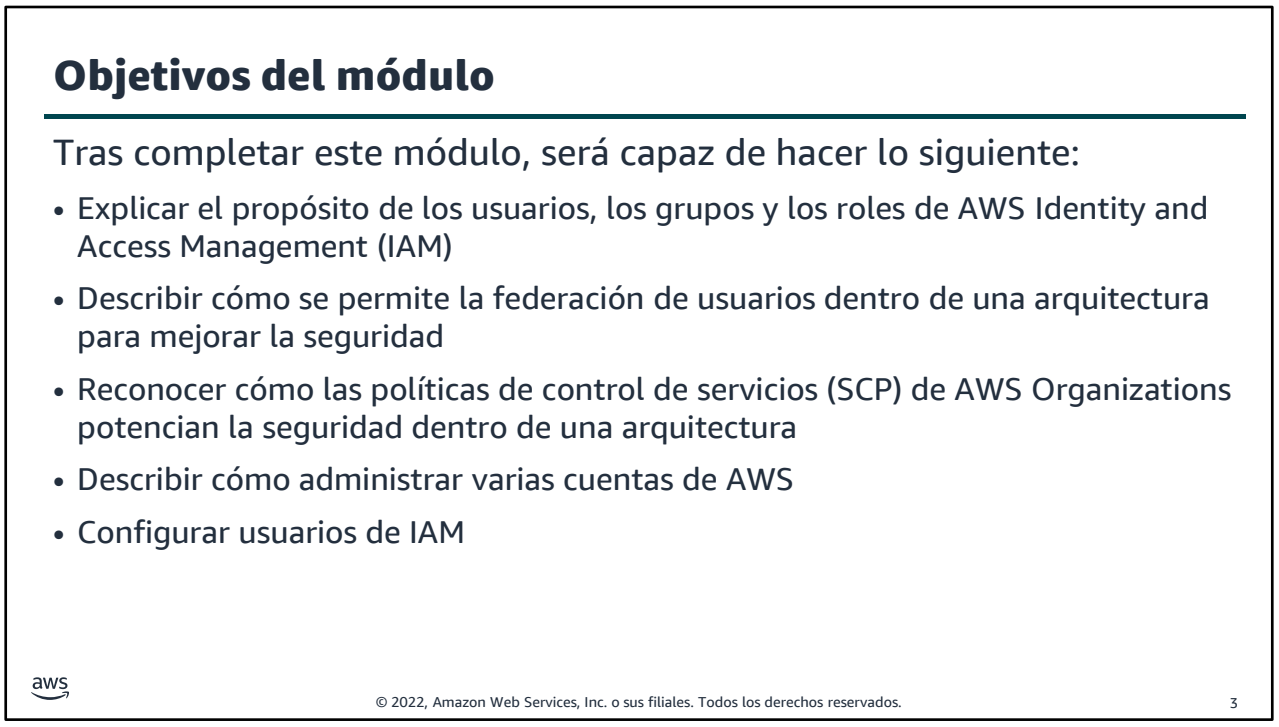
Bienvenido al Módulo 8: protección del acceso de los usuarios y las aplicaciones.



Este módulo contiene las siguientes secciones:
1. Necesidad de arquitectura
2. Usuarios de cuentas e IAM
3. Organización de usuarios
4. Federación de usuarios
5. Varias cuentas

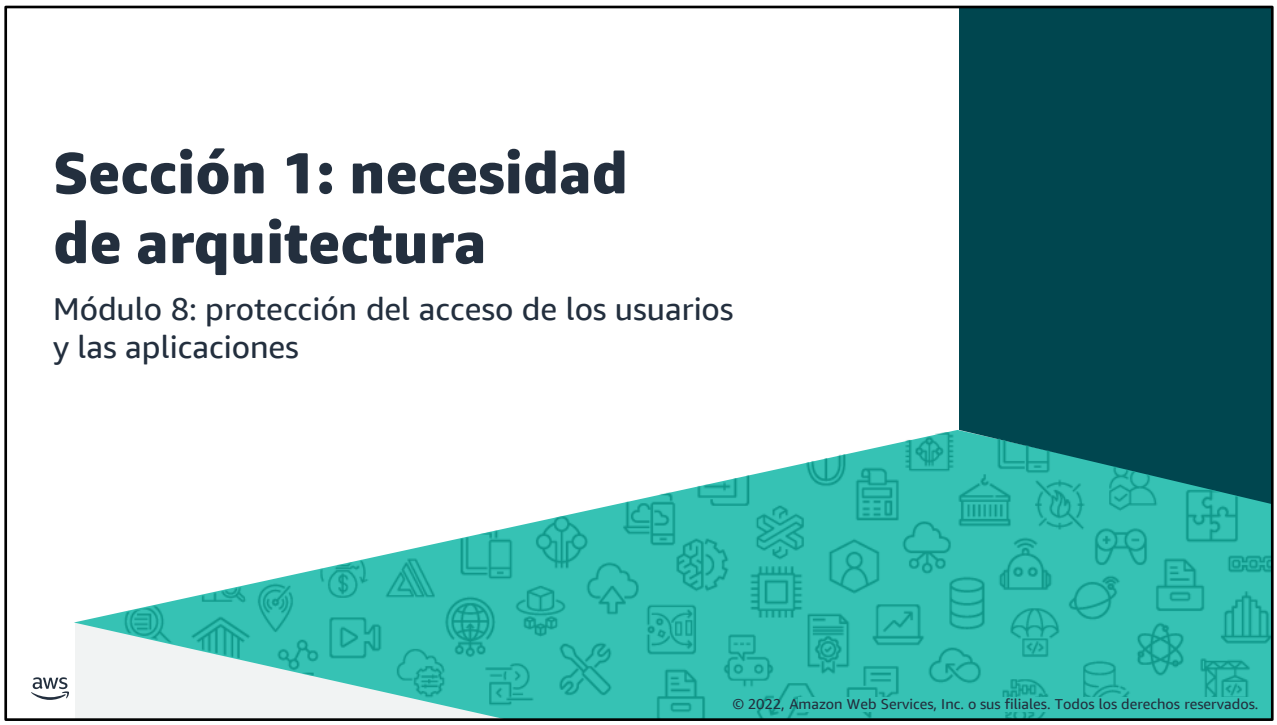
Este módulo también incluye:
- Unademostraciónquele indicará una función de uso común. Se adjunta un rol de IAM queotorga acceso a otros servicios de Amazon Web Services (AWS) a una instancia de Amazon Elastic Compute Cloud (Amazon EC2)
- Unaactividad que lo desafía a analizar documentos de políticas de AWS Identity and Access Management (IAM) para determinar qué acciones permiten o deniegan las políticas
- Unlaboratorio de desafíos donde se utiliza IAM para configurar usuarios, grupos y políticas de acceso que son adecuadas para el caso práctico de la cafetería

Finalmente, se le pedirá que complete una evaluación de conocimientos que pondrá a prueba su comprensión de los conceptos clave que se abordaron en este módulo.



Tras completar este módulo, será capaz de hacer lo siguiente:

- Explicar el propósito de los usuarios, los grupos y los roles de AWS Identity and Access Management (IAM).
- Describir cómo se permite la federación de usuarios dentro de una arquitectura para mejorar la seguridad
- Reconocer cómo las políticas de control de servicios (SCP) de AWS Organizations potencian la seguridad dentro de una arquitectura
- Describir cómo administrar varias cuentas de AWS
- Configurar usuarios de IAM



Introducción a la Sección 1: necesidad de arquitectura

Requisitos de la empresa de cafetería

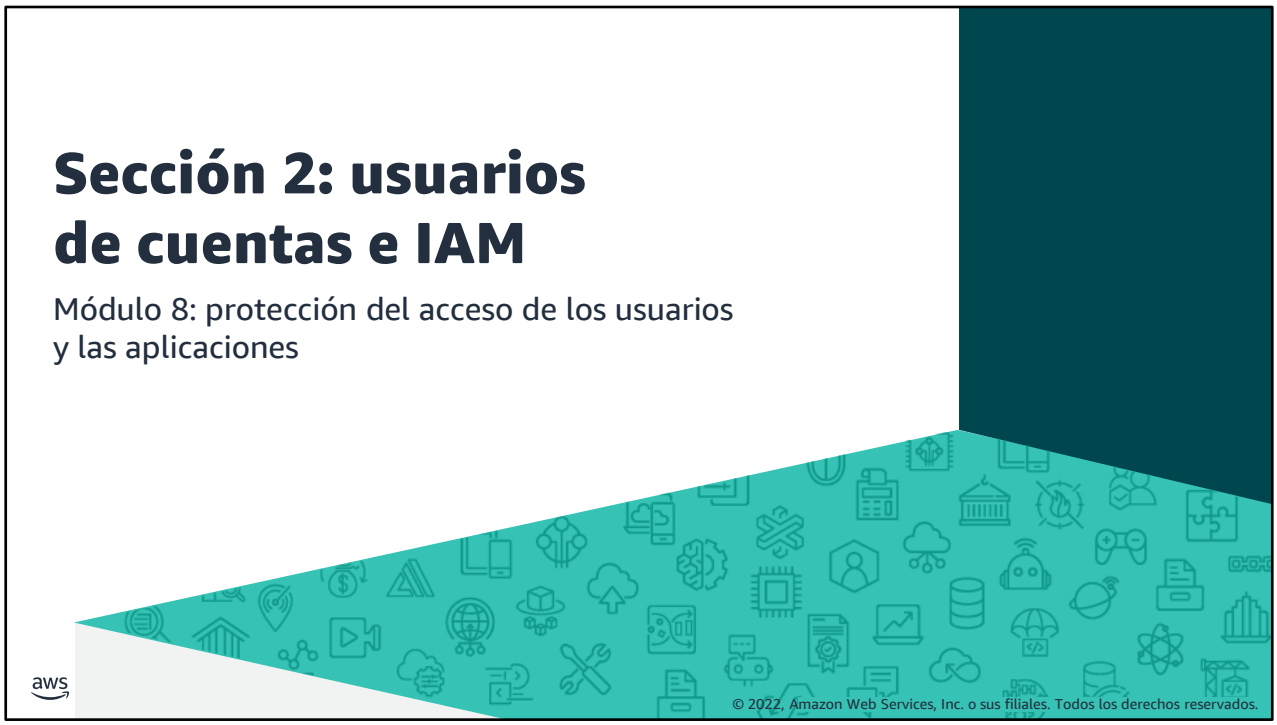
La cafetería debe definir qué nivel de acceso deben tener los usuarios y sistemas a través de los recursos de la nube y luego implementar estos controles de acceso en la cuenta de AWS.

5

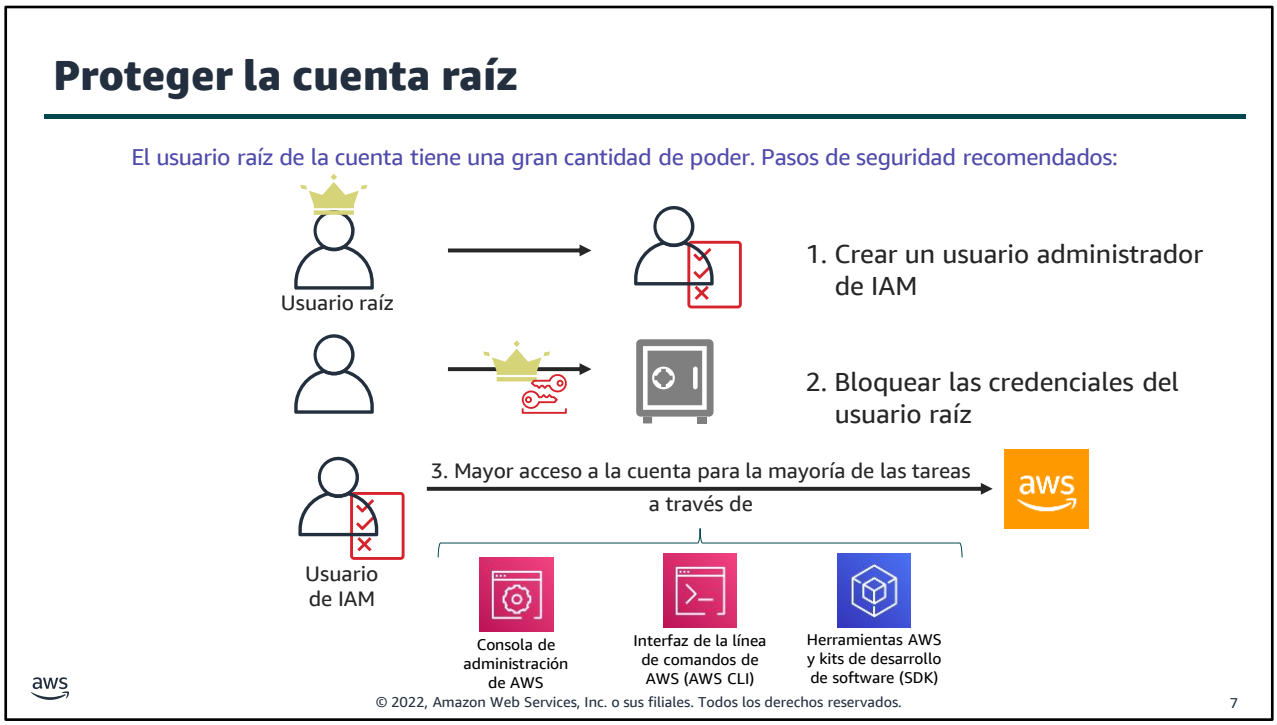
La cafetería debe definir qué nivel de acceso deben tener los usuarios y sistemas entre sus recursos de la nube. Luego, deben implementar estos controles de acceso en su cuenta de AWS.

La cafetería es ahora lo suficientemente grande como para que los miembros del equipo que crean, mantienen o acceden a aplicaciones en AWS se especialicen en roles (como desarrollador o administrador de bases de datos). Hasta ahora, no se habían esforzado por definir claramente qué nivel de acceso debería tener cada usuario en función de sus roles y responsabilidades.

A lo largo de este módulo, aprenderá acerca de IAM, que proporciona las funciones que necesita para cumplir con estos nuevos requisitos empresariales.



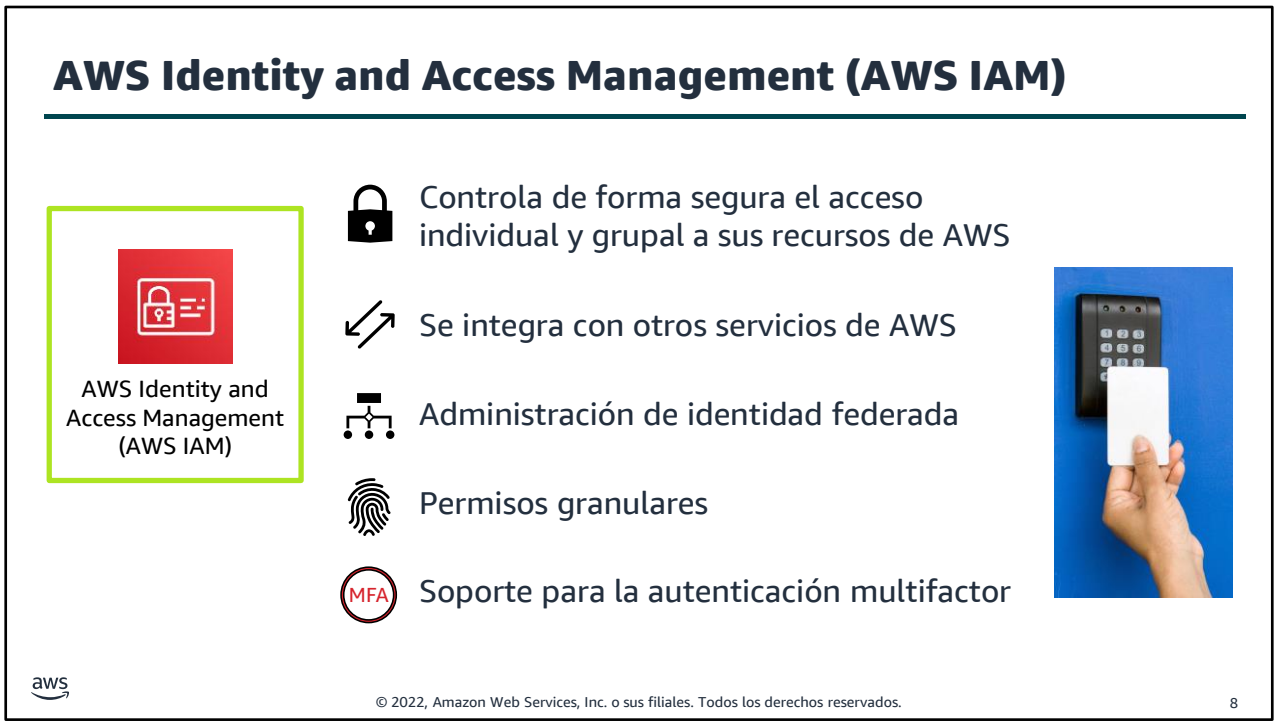
Introducción a la Sección 2: usuarios de cuentas e IAM.



Cuando crea una cuenta de AWS, comienza con un *usuario raíz*. Este usuario puede iniciar sesión en la Consola de administración de AWS con la dirección de correo electrónico que se utilizó para crear la cuenta.

El usuario raíz de la cuenta de AWS tiene acceso completo a todos los recursos en la cuenta, incluida la información de facturación, los datos personales en el perfil del usuario y todos los recursos que fueron creados en cualquier servicio de AWS de la cuenta. No puede controlar los privilegios de las credenciales de usuario raíz de la cuenta de AWS.

AWS recomienda enfáticamente que no utilice las credenciales del usuario raíz para las interacciones diarias con AWS. En cambio, cree uno o más usuarios de IAM. Guarde las credenciales del usuario raíz en una ubicación segura. Para la mayoría de las tareas de administración y el acceso a cuentas en curso, puede usar las credenciales de usuario de IAM.



AWS Identity and Access Management también se conoce como IAM. Es un servicio que permite configurar control de acceso detallado a los recursos de AWS. IAM habilita las prácticas recomendadas de seguridad al permitirle otorgar credenciales de seguridad únicas a los usuarios y grupos. Estas credenciales especifican a qué interfaces de programación de aplicaciones (API) de servicios de AWS y recursos pueden acceder. IAM es seguro según la configuración predeterminada. Los usuarios no tienen acceso a los recursos de AWS hasta que se les otorguen permisos explícitamente.

IAM se encuentra integrado en la mayoría de los servicios de AWS. Puede definir los controles de acceso desde un lugar en la consola de administración de AWS y surtirán efecto en todo su entorno de AWS.

Puede utilizar IAM para otorgar a sus empleados y aplicaciones acceso a la Consola de administración de AWS y a las API de servicios de AWS, utilizando sus sistemas de identidad existentes. AWS admite la federación de sistemas corporativos como Microsoft Active Directory y proveedores de identidad basados en estándares. IAM también admite la autenticación multifactor (MFA). Si MFA se encuentra habilitado y un usuario de IAM intenta iniciar sesión, se le solicitará un código de autenticación. El código de autenticación se entrega a un dispositivo MFA de AWS. El dispositivo MFA puede ser un dispositivo MFA de hardware. También puede ser un dispositivo MFA virtual que al que accede el usuario a través de una aplicación que se ejecuta en el smartphone del usuario como Google Authenticator.

Puede crear cuentas que tengan privilegios similares a los del usuario raíz de la cuenta de AWS. Sin embargo, es mejor crear cuentas administrativas que otorguen solo los permisos necesarios. Siga el principio de mínimo privilegio. Por ejemplo, pregúntese si su administrador de base de datos (DBA) debería poder aprovisionar instancias de EC2. Si la respuesta es no, entonces aprovisione las cuentas en consecuencia.

Componentes de IAM: revisión

Usuario de IAM

Definido en su cuenta de AWS. Utilice las credenciales para autenticar mediante programación o a través de la Consola de administración de AWS.

Grupo de IAM

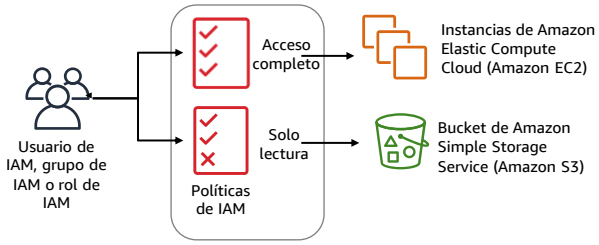
Un conjunto de usuarios de IAM a los que se les otorga una autorización idéntica.

Política de IAM

Define a qué recursos se puede acceder y el nivel de acceso a cada uno de estos recursos.

Rol de IAM

Mecanismos para otorgar acceso temporal para realizar solicitudes de servicios de AWS. Que puede ser *asumido* por una persona, aplicación o servicio.



aws

 9

Para comprender cómo utilizar IAM para proteger su cuenta de AWS, es importante entender el rol y la función de cada uno de los cuatro componentes de IAM.

Un*usuario de IAM*es una persona o aplicación que está definida en una cuenta de AWS y que debe realizar llamadas API a productos de AWS. Cada usuario debe tener un nombre único (sin espacios en el nombre) dentro de la cuenta de AWS y un conjunto de credenciales de seguridad que no se comparte con otros usuarios. Estas credenciales son diferentes de las credenciales de seguridad del usuario raíz de la cuenta de AWS. Cada usuario está definido en una y solo una cuenta de AWS.

Un *grupo de IAM* es un conjunto de usuarios de IAM. Puede utilizar grupos de IAM para simplificar la forma de especificar y administrar permisos para varios usuarios.

Una*política de IAM*es un documento que define permisos para determinar qué pueden y no pueden hacer los usuarios en la cuenta de AWS.

Un rol de IAM es una herramienta para otorgar acceso temporal a recursos de AWS específicos en una cuenta de AWS.

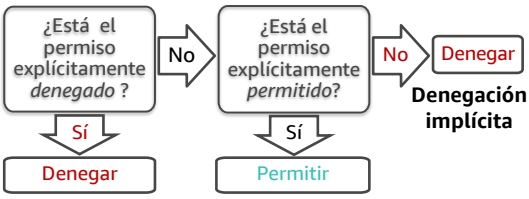
Permisos de IAM

Política de IAM

Los permisos se especifican en una política de IAM:

- Un documento tiene formato JavaScript Object Notation (JSON).
- Define qué recursos y operaciones están permitidos
- Práctica recomendada: siga el principio de mínimo privilegio
- Existen dos tipos de políticas:
  - Basada en la identidad: adjuntar a una entidad principal de IAM
  - Basada en recursos: adjuntar a un recurso de AWS

Cómo IAM determina los permisos en el momento de la solicitud:



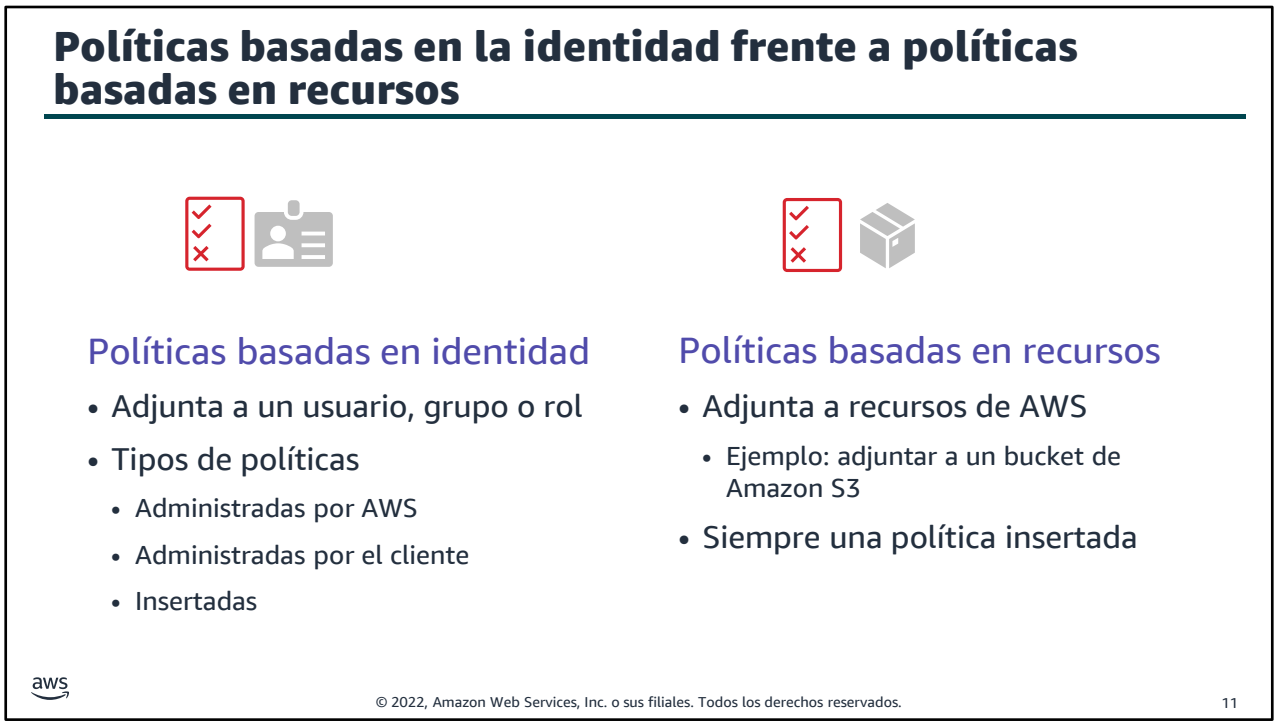
 10

EnIAM, los permisos se definen en los documentos de políticas de IAM. Las políticas le permiten ajustar los privilegios que se otorgan a las entidades principales. Las entidades principales de ejemplo son los usuarios de IAM, roles de IAM u otros servicios de AWS.

Cuando IAM determina si se autoriza un permiso, primero verifica la existencia de alguna *política de denegación explícita* aplicable. Si no existe una denegación explícita, comprueba si existe alguna *política de permiso explícito*. Si no existe una política de denegación explícita o de permiso explícito, IAM vuelve al valor predeterminado y deniega el acceso. Este proceso se denomina *denegación implícita*. Al usuario se le permitirá realizar la acción sólo si la acción solicitada *no* se deniega explícitamente y *se*permite explícitamente.

Cuando desarrolla políticas de IAM, puede resultar difícil determinar si se otorgará acceso a un recurso a una entidad de IAM. El Simulador de políticas de IAM es una herramienta útil para probar y solucionar problemas de políticas de IAM.

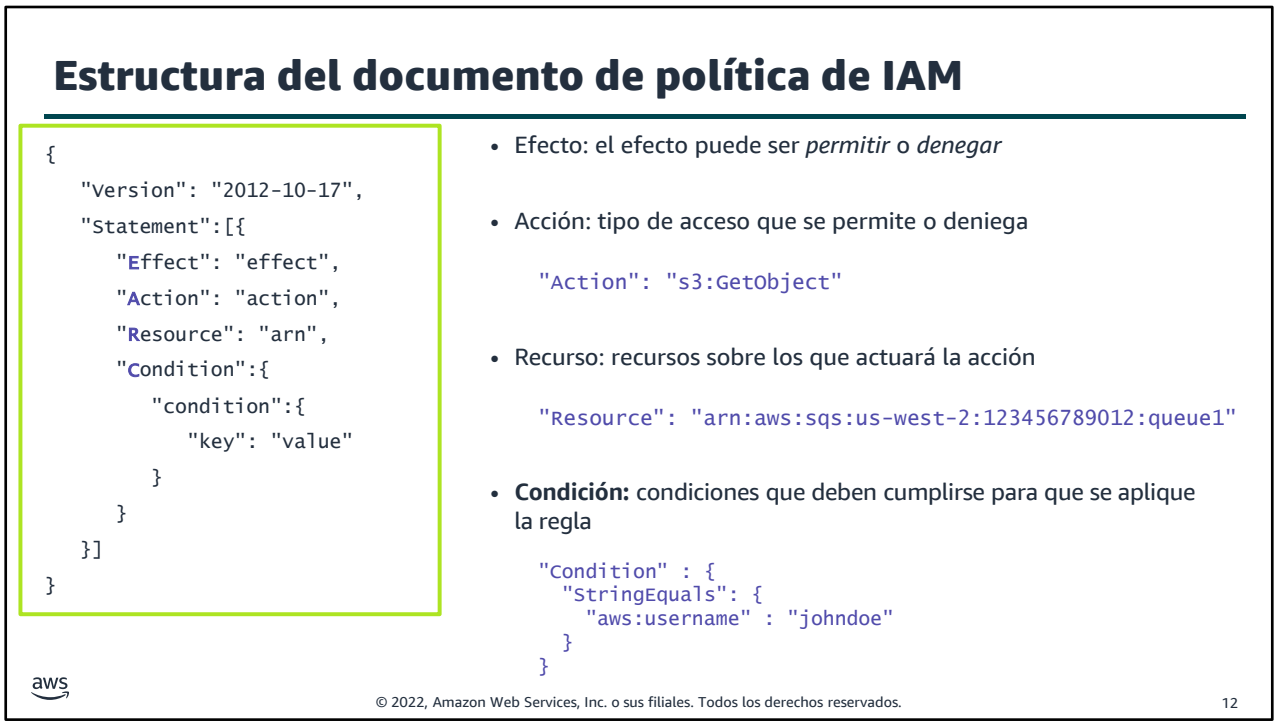
Las políticas se almacenan como documentos con notación de objetos JavaScript (JSON). Se adjuntan a las entidades principales como *políticas basadas en la identidad*, o a recursos como *políticas basadas en recursos*.



Las políticas basadas en la identidad son políticas de permisos que puede adjuntar a una entidad principal (o identidad), tal como un usuario, rol o grupo de IAM. Estas políticas c*ontrolan qué acciones puede realizar dicha identidad, en qué recursos y en qué condiciones*.

Las políticas basadas en la identidad pueden clasificarse a su vez como administradas por AWS, administradas por el cliente o insertadas. *Las* políticas administradas por AWS son creadas y administradas por AWS y puede adjuntarlas a varios usuarios, grupos y roles en su cuenta de AWS. Si es nuevo en el uso de las políticas, le recomendamos que comience utilizando las políticas administradas por AWS. *Las* políticas administradas por el cliente son aquellas que usted crea y administra en su cuenta de AWS. Las políticas administradas por el cliente proporcionan un control más preciso sobre sus políticas que las políticas administradas por AWS. Puede crear y editar una política de IAM en el editor visual o al crear un documento de política en formato JSON directamente. *Las* políticas insertadas son políticas que usted crea y administra, y que están insertadas directamente en un único usuario, grupo o rol.

Las políticas basadas en recursos son documentos de políticas JSON que adjunta a un recurso, tal como un bucket de Amazon Simple Storage Service (Amazon S3). Estas políticas controlan *qué acciones puede realizar una entidad principal especificada en dicho recurso y en qué condiciones.* Las políticas basadas en recursos son políticas insertadas y no hay políticas administradas basadas en recursos.



Las políticas de IAM se almacenan en AWS como documentos JSON. Las políticas basadas en la identidad son documentos de políticas que adjunta a un usuario o rol. Las políticas basadas en recursos son documentos de políticas que adjunta a un recurso. Un documento de política incluye uno o más enunciados individuales. Cada enunciado incluye información sobre un único permiso. Si una política incluye varios enunciados, AWS aplica un OR lógico entre enunciados cuando los evalúa.

Los siguientes son elementos comunes que se encuentran en un documento de políticas de IAM:

- **Versión**: especifique la versión del idioma de la política que desea utilizar. Como práctica recomendada, utilice la última versión del 17-10-2012.
- **Enunciado**: utilice este elemento de la política principal como un contenedor para los siguientes elementos. Puede incluir más de un enunciado en una política.
- **Efecto**: utilice Allow (Permitir) o Deny (Denegar) para indicar si la política permite o deniega el acceso.
- **Entidad principal**: si crea una política basada en recursos, debe indicar la cuenta, el usuario, el rol o el usuario federado al que desea permitir o denegar el acceso. Si va a crear una política de permisos de IAM para adjuntarla a un usuario o un rol, no puede incluir este elemento. La entidad principal está implícita como ese usuario o rol.
- **Acción**: incluye una lista de acciones que la política permite o deniega.
- **Recurso**: si crea una política de permisos de IAM, debe especificar una lista de recursos a los que se aplican las acciones. Si crea una política basada en recursos, este elemento es opcional.
- **Condición (Opcional)**: especifica las circunstancias en las cuales la política otorga permisos.

ARN y comodines

- Los recursos se identifican usando el formato Amazon Resource Name (ARN)
  - Sintaxis: `arn:partition:service:region:account:resource`
  - Ejemplo: `"Resource": "arn:aws:iam::123456789012:user/mmajor"`
- Puede utilizar un comodín (*) para brindar acceso a todas las acciones para un servicio específico de AWS.
  - Ejemplos:
    - `"Action": "s3:*"`
    - `"Action": "iam:*AccessKey*"`

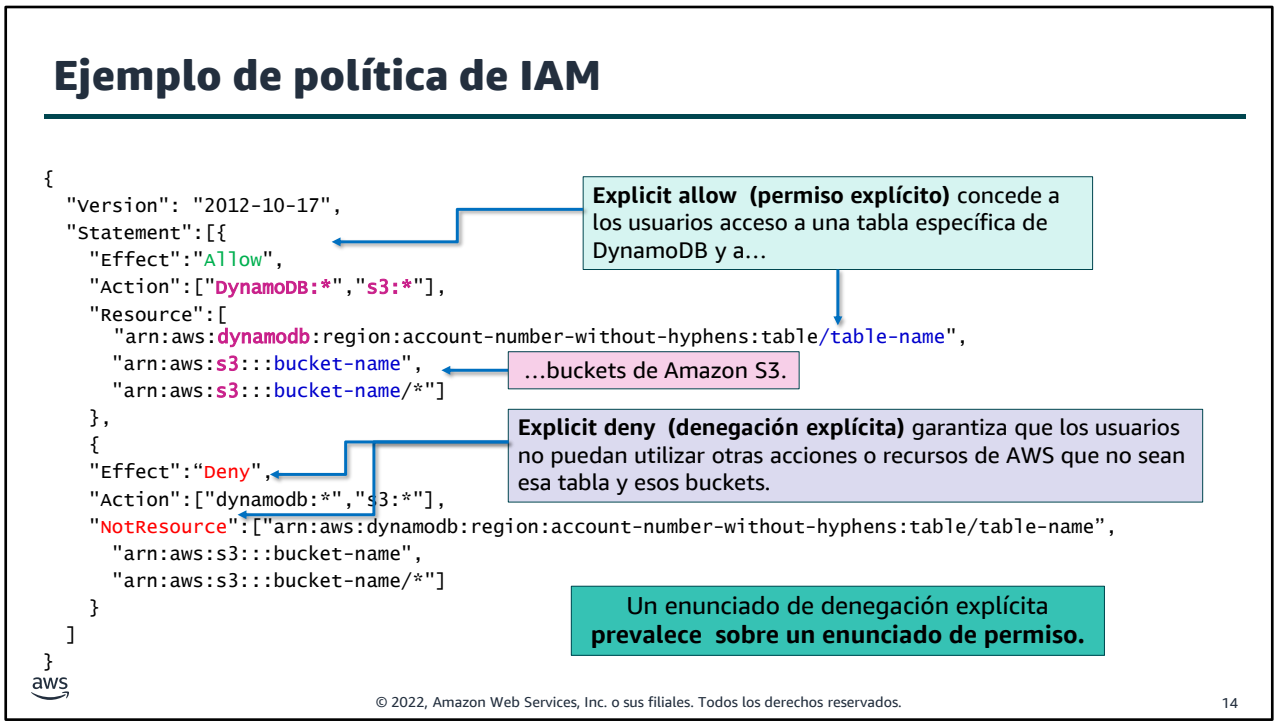
13

Para las políticas basadas en la identidad (permisos de IAM), debe especificar una lista de recursos a los que se aplican las acciones. El *elemento* Resource especifica el objeto o los objetos que cubre el enunciado. Los enunciados deben incluir un elemento Resource o NotResource.

La mayoría de los recursos tienen un nombre descriptivo (por ejemplo un usuario llamado *Bob*o un grupo llamado*Desarrolladores).* Sin embargo, el lenguaje de la política de permisos requiere que especifique el recurso o recursos a través del siguiente formato de *Amazon Resource Name* (ARN).

Cada servicio tiene su propio conjunto de recursos. Aunque siempre utiliza un ARN para especificar un recurso, los detalles del ARN de un recurso dependen del servicio y del recurso. Para obtener información sobre cómo especificar un recurso, consulte la documentación del servicio para cuyos recursos está escribiendo un enunciado.

También puede utilizar comodines en documentos de políticas de IAM, como en ARN o en Acciones. Puede utilizar el carácter comodín (*). Un asterisco (*) representa cualquier combinación de cero o más caracteres. Por ejemplo, un valor de *"Acción"* de *"s3:*"* se aplica a todas las acciones S3. También puede utilizar comodines (*) como parte del nombre de la acción. Por ejemplo, el valor de la *"Acción"* de *"iam:*AccessKey*"* se aplica a todas las acciones de IAM que incluyen la cadena *AccessKey*, incluidas CreateAccessKey, DeleteAccessKey, ListAccessKeys, y UpdateAccessKey.



Como se mencionó previamente, los documentos de políticas de IAM están escritos en JSON.

Esta política de IAM de ejemplo otorga acceso de usuario solo a los siguientes recursos:
- La tabla de Amazon DynamoDB cuyo nombre está representado por *table-name*.
- El bucket de S3 de la cuenta de AWS, cuyo nombre está representado por*bucket-name*y todos los objetos que contiene.

La política de IAM también incluye un elemento de denegación explícita ("Effect":"Deny"). El elemento *NotResource* ayuda a garantizar que los usuarios no puedan usar ninguna acción o recurso de DynamoDB o S3, salvo los especificados en la política. Este es el caso aunque se hayan concedido permisos en otra política. Un enunciado de denegación explícita prevalece sobre un enunciado de permiso.

Actividad: análisis de las políticas de IAM

Foto de Pixabay de Pexels.

 15

En esta actividad dirigida por el instructor, se presentarán ejemplos de políticas de IAM. Para cada política, se le harán preguntas acerca de si la política permite o deniega acciones particulares. El instructor lo dirigirá a un análisis de cada pregunta y revelará las respuestas correctas una a la vez.

Actividad: análisis de las políticas (1 de 3)

Considere esta política de IAM y luego responda las preguntas.

```
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": [
         "iam:Get*",
         "iam:List*"
      ],
      "Resource": "*"
   }
}
```

1. ¿A qué servicio de AWS le otorga acceso esta política?
2. ¿Le permite crear un usuario, grupo, política o rol de IAM?
3. Vaya a https://docs.aws.amazon.com/IAM/latest/UserGuide/ *y en el panel de navegación izquierdo expanda Reference > Policy Reference > Actions, Resources, and Condition Keys.* Seleccione *Identity and Access Management.* Desplácese hasta la lista Actions Defined by Identity And Access Management .

   Nombre al menos tres acciones específicas que permite la acción iam:Get*

aws

16

Mire el documento de política de IAM de ejemplo. El instructor ahora le hará una serie de preguntas para evaluar si comprende qué acciones permitirá y denegará esta política.

Actividad: análisis de la política de IAM (1 de 3) - Respuestas

Considere esta política de IAM y luego responda las preguntas.

```
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": [
         "iam:Get*",
         "iam:List*"
      ],
      "Resource": "*"
   }
}
```

1. ¿A qué servicio de AWS le otorga acceso esta política?
   - RESPUESTA: El servicio de IAM.
2. ¿Le permite crear un usuario, grupo, política o rol de IAM?
   - RESPUESTA: No. El acceso está limitado a solicitudes *get y list*. Otorga efectivamente permisos de solo lectura.
3. Vaya a https://docs.aws.amazon.com/IAM/latest/UserGuide/ y en el panel de navegación izquierdo expanda *Reference > Policy Reference > Actions, Resources, and Condition Keys.* Seleccione *Identity and Access Management.* Desplácese hasta la lista Actions Defined by Identity And Access Management.

   Nombre al menos tres acciones específicas que permite la acción iam:Get*
   - RESPUESTA: iam:Get* permite muchas acciones específicas, incluidas GetGroup, GetPolicy, GetRole, y otras.

aws

 17

Se revelan las respuestas.

Actividad: análisis de la política de IAM (2 de 3)

Considere esta política de IAM y luego responda las preguntas.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

1. ¿Le permite la política terminar cualquier instancia EC2 en cualquier momento sin condiciones?
2. ¿Se le permite hacer la llamada de terminación de instancia desde cualquier lugar?
3. ¿Puede terminar instancias si realiza la llamada desde un servidor que tiene asignada una dirección IP de *192.0.2.243*?

18

Analice el segundo ejemplo de archivo de política de IAM. La primera parte muestra Effect: Allow and Action ec2:TerminateInstance for resource. La segunda parte muestra effect Deny for action ec2:TerminateInstances with condition NotIpAddress aws:SourceIp 192.0.2.0/24 and 203.0.113.0/24 for resource. El instructor ahora le volverá a hacer una serie de preguntas para evaluar si comprende qué acciones permitirá y denegará esta política.

Actividad: análisis de la política de IAM (2 de 3) - Respuestas

Considere esta política de IAM y luego responda las preguntas según se presentan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

1. ¿Le permite la política terminar cualquier instancia EC2 en cualquier momento sin condiciones?
   - RESPUESTA: No. El primer objeto del enunciado lo permite. Sin embargo, el segundo objeto del enunciado aplica una condición.
2. ¿Se le permite hacer la llamada de terminación de instancia desde cualquier lugar?
   - RESPUESTA: No. Solo puede realizar la solicitud desde uno de los dos rangos de direcciones IP que se especifican en *aws:SourceIp.*
3. ¿Puede terminar instancias si realiza la llamada desde un servidor que tiene asignada una dirección IP de *192.0.2.243*?
   - RESPUESTA: Sí, porque el rango de direcciones IP de enrutamiento entre dominios sin clase (CIDR) 192.0.2.0/24 incluye las direcciones IP 192.0.2.0 a 192.0.2.255. Se puede utilizar un recurso como la herramienta CIDR to IP Range para calcular el rango de un bloque de CIDR.

19

Se revelan las respuestas.

Para accesibilidad: documento de política de ejemplo en formato JSON. Muestra una sección del enunciado con dos partes. La primera parte muestra Effect:Allow and Action EC2:TerminateInstance for resource *. La segunda parte muestra effect Deny for action EC2:TerminateInstances with condition NotIpAddress aws:SourceIp 192.0.2.0/24 and 203.0.113.0/24 for resource *. **Fin de la descripción de accesibilidad.**

Actividad: análisis de la política de IAM (3 de 3)

Considere esta política de IAM y luego responda las preguntas.

```
{
  "Version": "2012-10-17",
  "Statement":[{
      "Condition": {
        "StringNotEquals": {
           "ec2:InstanceType": [
              "t2.micro",
              "t2.small"
           ]
        }
      },
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Action": [
         "ec2:RunInstances",
         "ec2:StartInstances"
      ],
     "Effect": "Deny"
   }
]
```

1. ¿Qué acciones permite la política?
2. Supongamos que la política incluye un objeto de enunciado adicional, como este ejemplo:

```
{
     "Effect": "Allow",
     "Action": "ec2:*",
     "Resource": "*"
}
```

   ¿Cómo restringiría la política el acceso que le otorga este enunciado adicional?

3. Si la política incluyera tanto el enunciado de la izquierda como el enunciado en la pregunta 2, ¿podría terminar una instancia m3.xlarge que existía en la cuenta?

aws

 20

Observe el tercer y último ejemplo de documento de política de IAM. El instructor ahora le volverá a hacer una serie de preguntas para evaluar si comprende qué acciones permitirá y denegará esta política.

Actividad: análisis de la política de IAM (3 de 3)

Considere esta política de IAM y luego responda las preguntas.

```
{
  "Version": "2012-10-17",
  "Statement":[{
      "Condition": {
        "StringNotEquals": {
           "ec2:InstanceType": [
              "t2.micro",
              "t2.small"
           ]
        }
      },
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Action": [
         "ec2:RunInstances",
         "ec2:StartInstances"
      ],
     "Effect": "Deny"
   }
]
```

1. ¿Qué acciones permite la política?
   - **RESPUESTA:** No le permite hacer nada *(el efecto es* Denegar).
2. Supongamos que la política incluye un objeto de enunciado adicional, como este ejemplo:

```
{
     "Effect": "Allow",
     "Action": "ec2:*",
     "Resource": "*"
}
```

   ¿Cómo restringiría la política el acceso que le otorga este enunciado adicional?
   - **RESPUESTA:** Tendría acceso completo al servicio Amazon EC2. Sin embargo, solo se le permitiría lanzar o iniciar instancias EC2 del tipo de instancia t2.micro o t2.small.
3. Si la política incluyera tanto el enunciado de la izquierda como el enunciado en la pregunta 2, ¿podría terminar una instancia m3.xlarge que existía en la cuenta?
   - **RESPUESTA:** Sí.

aws

21

Se revelan las respuestas.

AWS CloudTrail

AWS CloudTrail

- Registra y supervisa la actividad del usuario
- Proporciona un historial de eventos de la cuenta de AWS
  - Acciones realizadas a través de la Consola de administración de AWS, SDK, AWS CLI
  - Aumenta la visibilidad de la actividad de sus usuarios y recursos
  - Historial de eventos de 90 días proporcionado de forma predeterminada, sin costo
- Identificar
  - *Quién* accedió a su cuenta
  - *Cuándo* y desde *dónde*
  - *Qué* acción tomaron en un servicio de AWS
- Herramienta útil para
  - Realizar análisis de la seguridad
  - Descubrir qué llamadas fueron bloqueadas (por ejemplo, por políticas de IAM)

 22

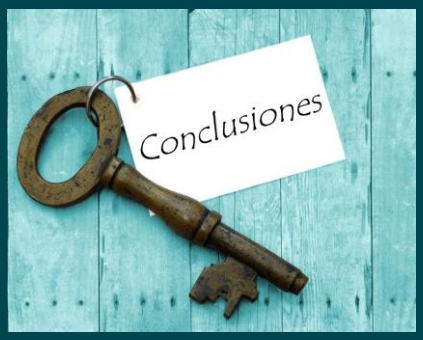
AWS CloudTrail es un servicio que permite la gobernanza, el cumplimiento y la auditoría de su cuenta de AWS. Con CloudTrail, puede monitorear de manera continua y retener la actividad de la cuenta que está relacionada con acciones en toda su infraestructura de AWS. Proporciona un historial de eventos de la actividad de la cuenta, incluidas las acciones realizadas a través de la Consola de administración de AWS, los SDK de AWS y las herramientas de línea de comandos. Con este historial de eventos se simplifican el análisis de seguridad, el seguimiento de cambios de recursos y la solución de problemas.

Puede descubrir y solucionar problemas operativos y de seguridad mediante la captura de un historial completo de los cambios que tuvieron lugar en su cuenta de AWS durante un período específico. Puede identificar qué usuarios y cuentas realizaron llamadas a AWS, cuándo se hicieron y cuáles fueron las direcciones IP de origen. CloudTrail le permite rastrear y responder automáticamente a la actividad de la cuenta que amenaza la seguridad de sus recursos de AWS.

Con la integración de Amazon EventBridge (se denominaba anteriormente Amazon CloudWatch Events), puede definir flujos de trabajo que se ejecutan cuando detecta eventos que pueden provocar vulnerabilidades de seguridad. Por ejemplo, puede crear un flujo de trabajo para agregar una política específica a un bucket de S3 cuando CloudTrail registra una llamada API que hace público ese bucket.

CloudTrail registra información importante sobre cada acción, incluido quién realizó la solicitud, los servicios utilizados, las acciones realizadas, los parámetros de las acciones y los elementos de respuesta que devolvió el servicio de AWS. El servicio también ayuda a las organizaciones a cumplir con los requisitos de cumplimiento y auditoría que deben respetar.

Conclusiones importantes de la Sección 2

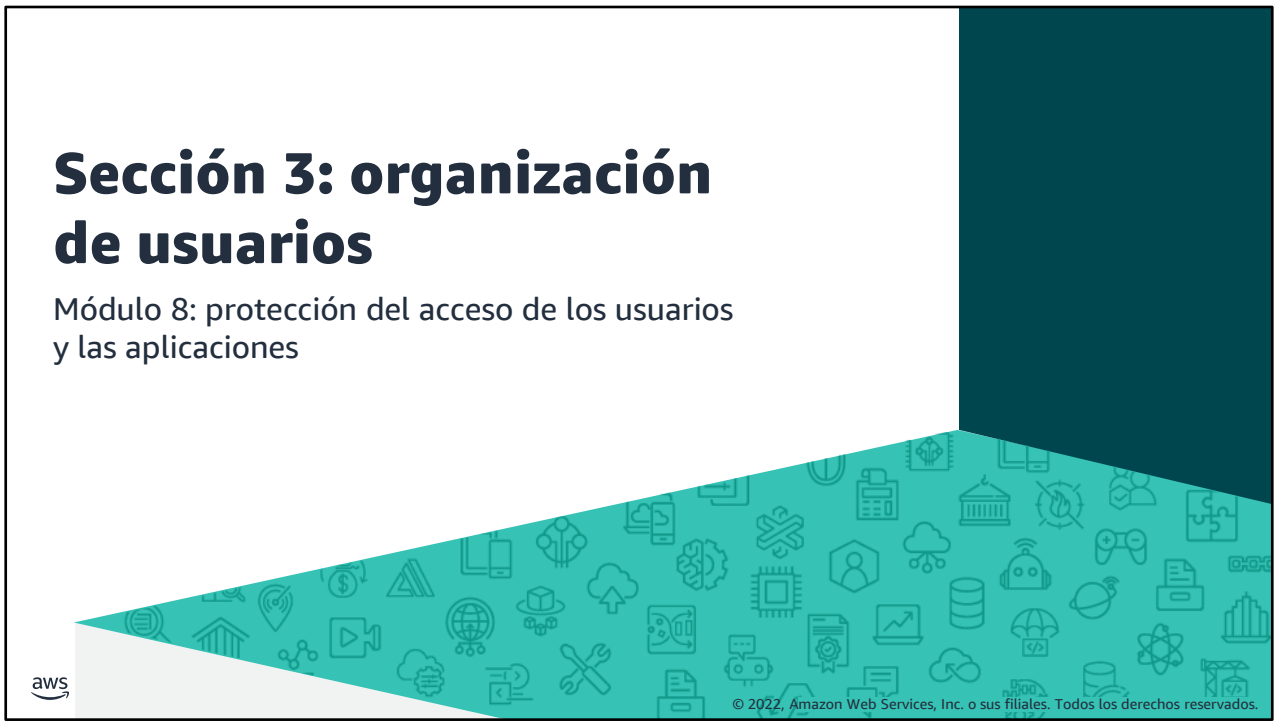


- Evite usar el usuario raíz de la cuenta para tareas comunes. En cambio, cree y utilice las credenciales de usuario de IAM.
- Los permisos para acceder a los recursos de la cuenta de AWS se definen en uno o más documentos de la política de IAM.
  - Adjuntar políticas de IAM a usuarios, grupos o roles de IAM.
- Cuando IAM determina los permisos, una denegación explícita siempre anulará cualquier enunciado de permiso.
- Es una práctica recomendada seguir el principio de mínimo privilegio cuando conceda acceso.

 23

Entre los aprendizajes clave de esta sección del módulo, se incluyen los siguientes:

- Evite usar el usuario raíz de la cuenta para tareas comunes. En cambio, cree y utilice las credenciales de usuario de IAM.
- Los permisos para acceder a los recursos de la cuenta de AWS se definen en uno o más documentos de la política de IAM.
  - Adjuntar políticas de IAM a usuarios, grupos o roles de IAM.
- Cuando IAM determina los permisos, una denegación explícita siempre anulará cualquier enunciado de permiso.
- Es una práctica recomendada seguir el principio de mínimo privilegio cuando conceda acceso a la cuenta.



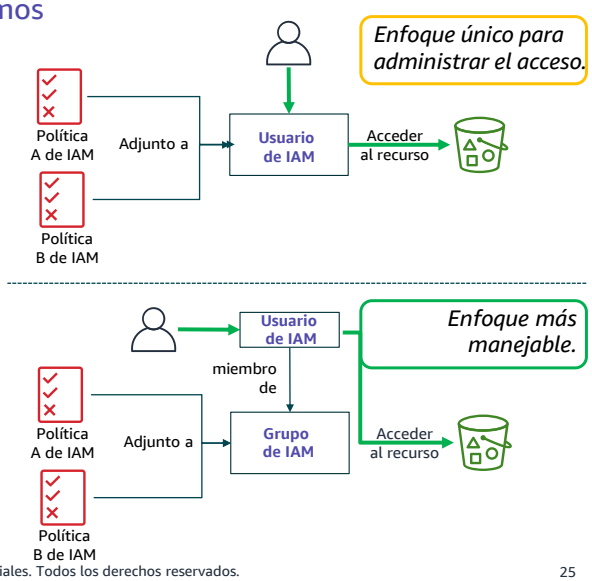
Introducción a la Sección 3: organización de usuarios

Grupos de IAM

Utilice los grupos de IAM para otorgar los mismos derechos de acceso para varios usuarios.

- Todos los usuarios de un grupo heredan los permisos que asignó al grupo
  - Facilita la administración del acceso entre múltiples usuarios.

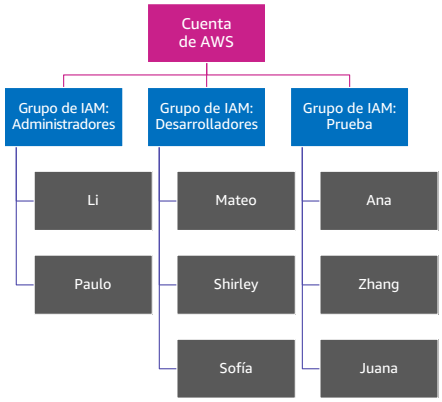
Sugerencia: Combine enfoques para un acceso individual detallado
- Agregue el usuario a un grupo para aplicar el acceso estándar según la función de trabajo
- Opcionalmente adjunte una política adicional al usuario para las excepciones necesarias



Un grupo de IAM es un conjunto de usuarios de IAM. Los grupos son una comodidad que facilita la administración de permisos para una serie de usuarios, en lugar de tener que administrar los permisos para cada usuario individual.

Administre la membresía del grupo como una lista simple:
- Agregue usuarios a un grupo o elimínelos de un grupo.
- Un usuario puede pertenecer a varios grupos.
- Los grupos no pueden pertenecer a otros grupos.
- A los grupos se les pueden otorgar permisos mediante políticas de control de acceso.
- Los grupos no tienen credenciales de seguridad y no pueden acceder a los servicios web directamente. Existen únicamente para facilitar la administración de los permisos de usuarios.

Ejemplo de grupos de IAM



Sugerencia: Cree grupos que reflejen las funciones de trabajo

- Si se contrata a un nuevo desarrollador, agréguelo al grupo *Desarrollador*
  - Hereda de inmediato el mismo acceso otorgado a otros desarrolladores
- Si Ana asume el nuevo rol de desarrolladora -
  - Elimínela del grupo *Prueba*
  - Agréguela al grupo *Desarrollador*
- Los usuarios pueden pertenecer a más de un grupo
  - Sin embargo, se aplicará la política más restrictiva

aws

 26

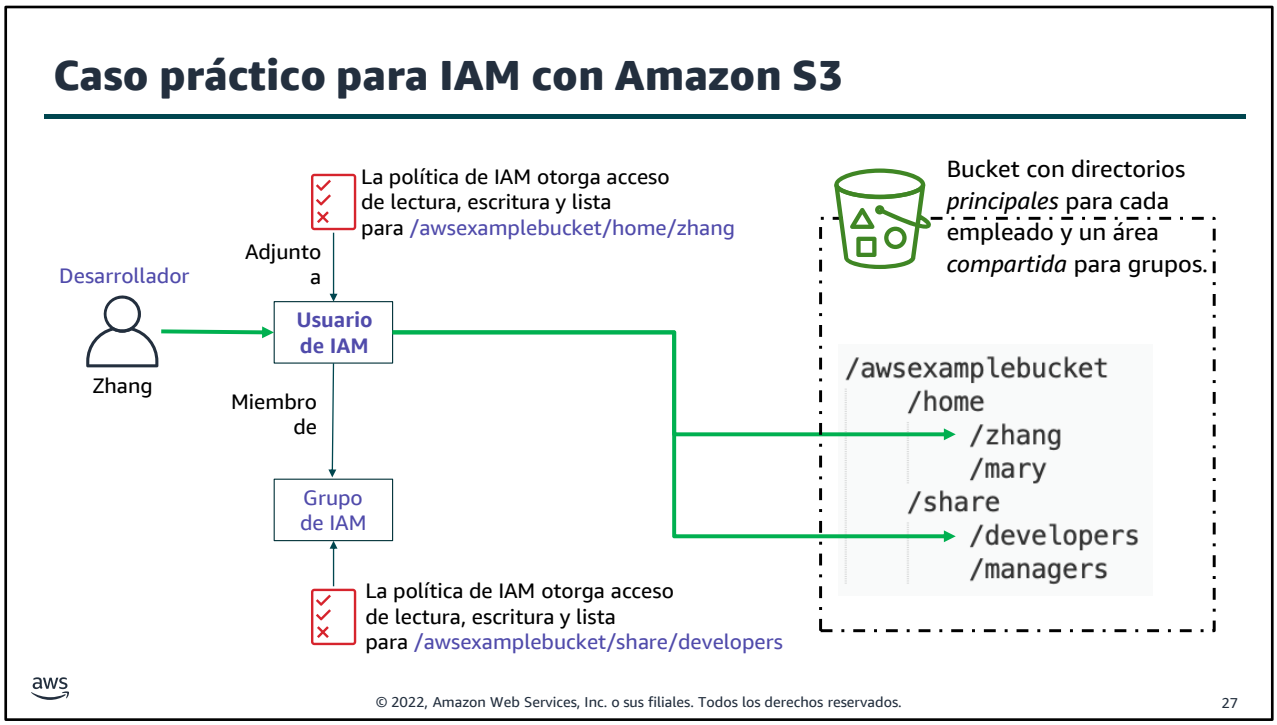
Por lo general, deseará crear grupos que reflejen las funciones de trabajo. Por ejemplo, podría crear un grupo para administradores, otro grupo para desarrolladores y otro grupo más para el equipo que realiza funciones de prueba.

Luego, adjunta uno o más archivos de políticas a cada grupo y agrega usuarios a los grupos. Los usuarios tienen los derechos de acceso asignados al grupo o grupos en los que se encuentran debido a su membresía en el grupo.

Si se contrata a un nuevo desarrollador, puede agregarlo al grupo de desarrolladores existente. Obtendrán el mismo acceso que ya tienen los demás desarrolladores.

Si una persona como Ana (que se muestra en el ejemplo) asume un nuevo rol en la organización, puede eliminarla del grupo *Prueba* y agregarla la grupo *Desarrolladores*. O si Ana realizará ambas funciones, puede dejarla en el grupo *Prueba* y agregarla al grupo*Desarrolladores*.

Si descubre que los desarrolladores necesitan acceso a algún recurso adicional en la cuenta, puede actualizar o agregar una política al grupo *Desarrolladores*. Todos los miembros del grupo obtendrán ese nivel adicional de acceso. Los grupos facilitan el mantenimiento de derechos de acceso consistentes entre los equipos.



Este ejemplo demuestra cómo se pueden configurar los permisos de IAM en un bucket de S3.

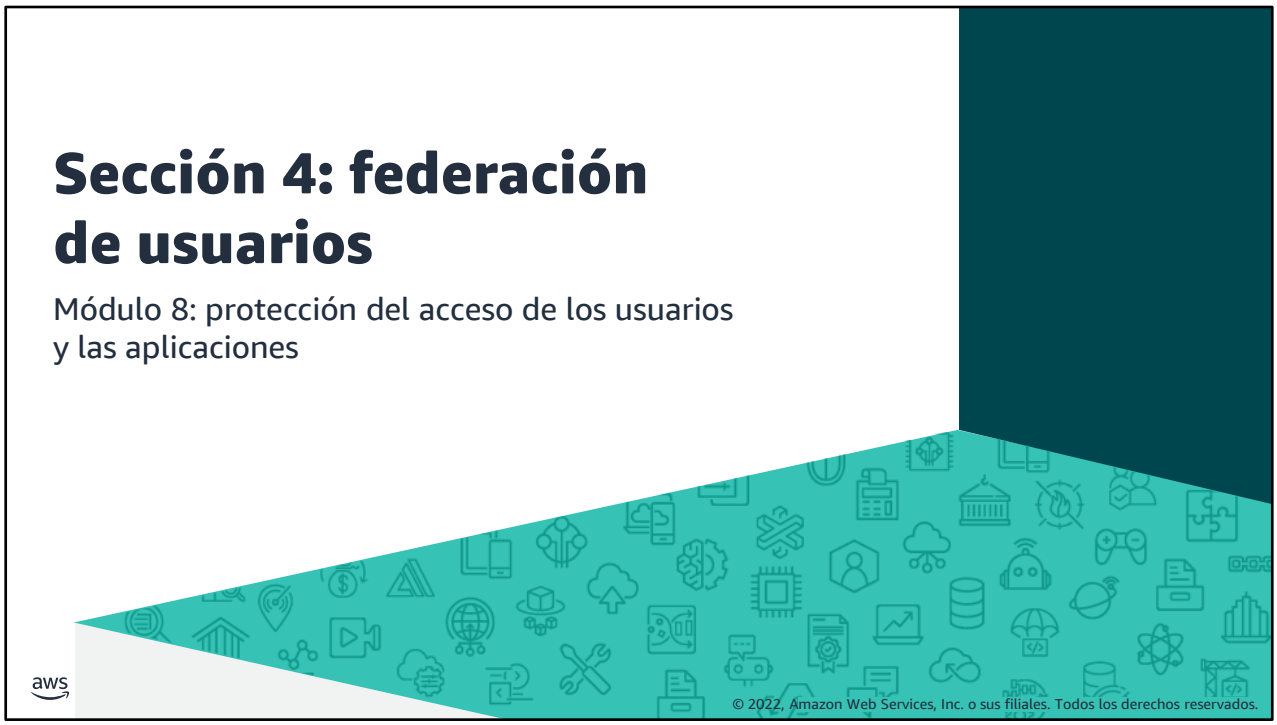
El*awsexamplebucket* tiene dos directorios principales. El directorio `principal` tiene subdirectorios para cada usuario, donde pueden almacenar trabajos individuales. El directorio `compartido` tiene subdirectorios donde diferentes equipos pueden almacenar contenido.

Si un nuevo miembro del equipo,*zhang*, se une a la organización como desarrollador, puede realizar tres acciones para otorgarle el acceso adecuado.

Primero, agregue a*zhang* a grupo de IAM para desarrolladores. Tenga en cuenta que este grupo tiene una política de IAM adjunta que otorga acceso a `/awsexamplebucket/share/developers`.

Luego, cree el directorio `/awsexamplebucket/home/zhang` en Amazon S3.

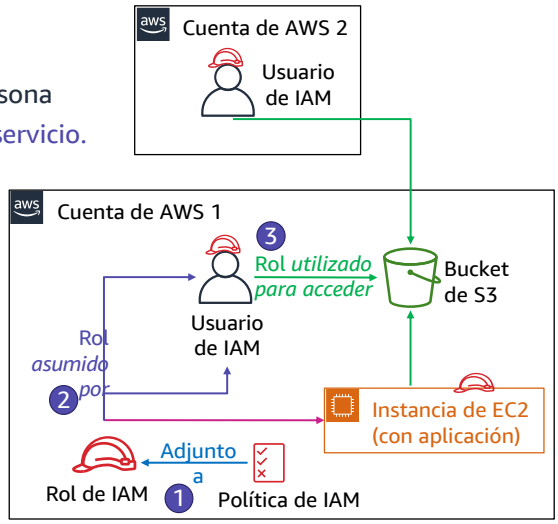
Finalmente, adjunte la política de IAM que otorga acceso al directorio`/awsexamplebucket/home/zhang` *directamente al usuario de IAM* zhang. El acceso de Zhang incluirá tanto los derechos otorgados por el grupo como también los derechos directamente vinculados al usuario principal de IAM.



Introducción a la Sección 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)



 29

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.

Demostración: Perfil de instancia EC2

 30

Ahora, el instructor podría optar por demostrar cómo adjuntar un rol de IAM a una instancia de EC2. Este rol otorga acceso a recursos de AWS a una aplicación.

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

 32

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.

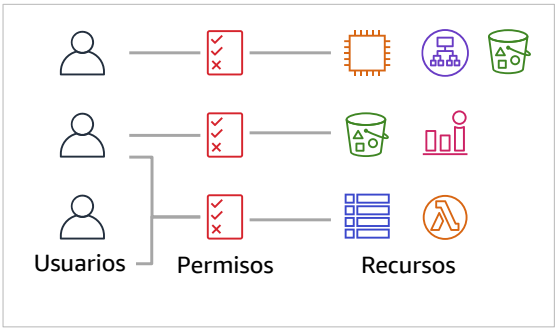
Cuando la operación AssumeRole de la API de AWS STS se invoca correctamente, el servicio web devuelve las credenciales temporales con privilegios limitados que solicitó el usuario de IAM o el usuario que se autenticó a través de federación. Por lo general, la operación AssumeRole se utiliza para acceso 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 recurso (puede llevar mucho tiempo seguir actualizando las políticas)



aws

 33

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.

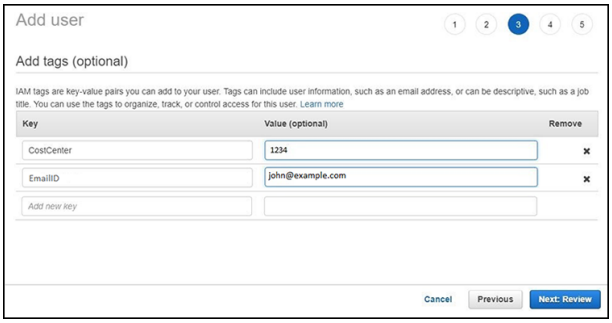
Históricamente, RBAC se ha utilizado en las instalaciones y en la nube. Con este modelo, otorga a los usuarios acceso explícito a un conjunto de permisos. Supongamos que tiene administradores de bases de datos, administradores de red y desarrolladores. Si tiene uno o más administradores de red que también 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 = unicornio
  - Pila = dev

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



 34

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.

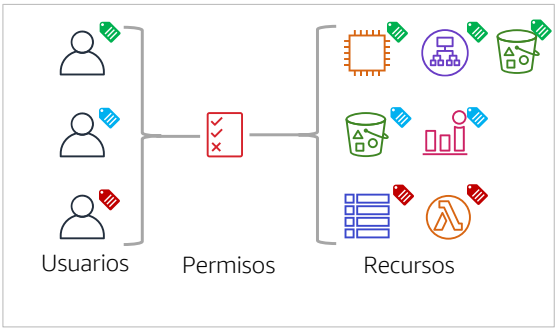
Las etiquetas tienen muchos usos prácticos. Por ejemplo, puede crear *etiquetas técnicas* para identificar que un recurso es un servidor web, parte de un proyecto específico, parte de un entorno específico (prueba, desarrollo o producción), entre otros. También puede crear *etiquetas empresariales* para 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



aws

35

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

aws

 36

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

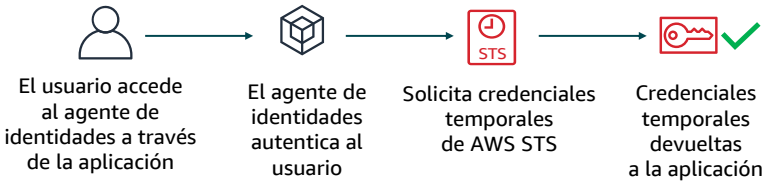
Federación de identidades

- 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

Descripción general de autenticación de IdP



aws

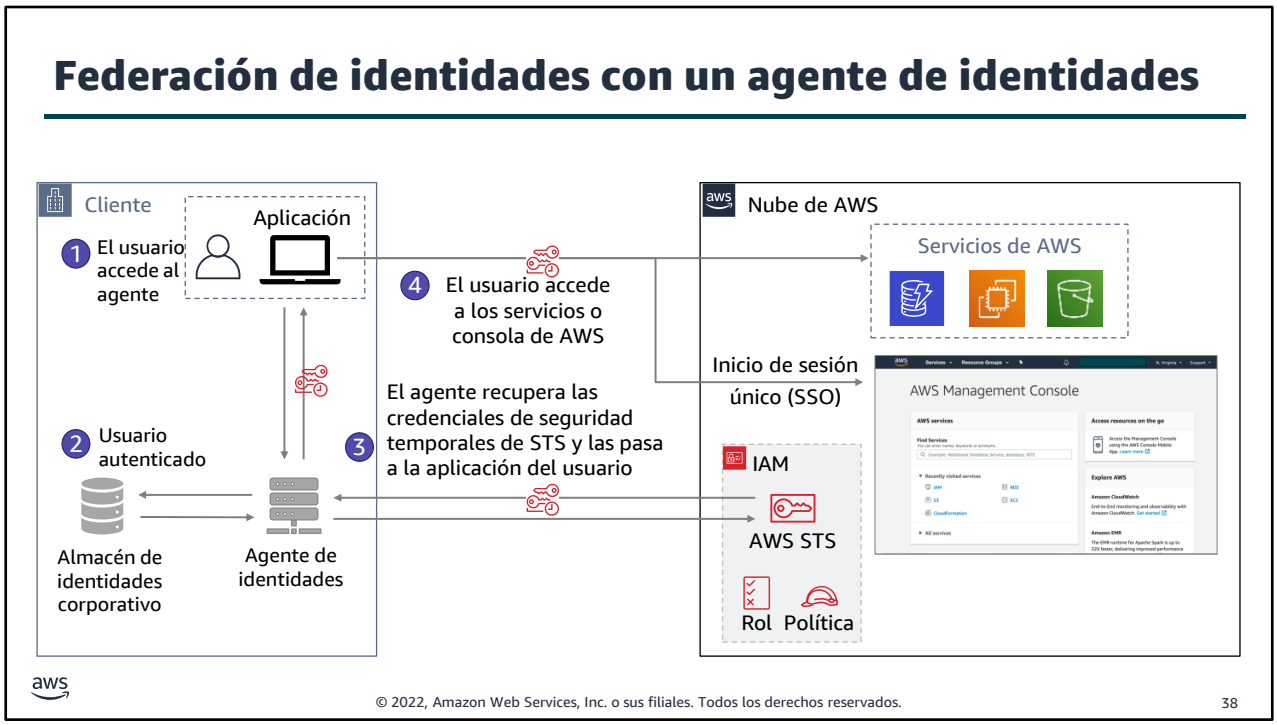
 37

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.

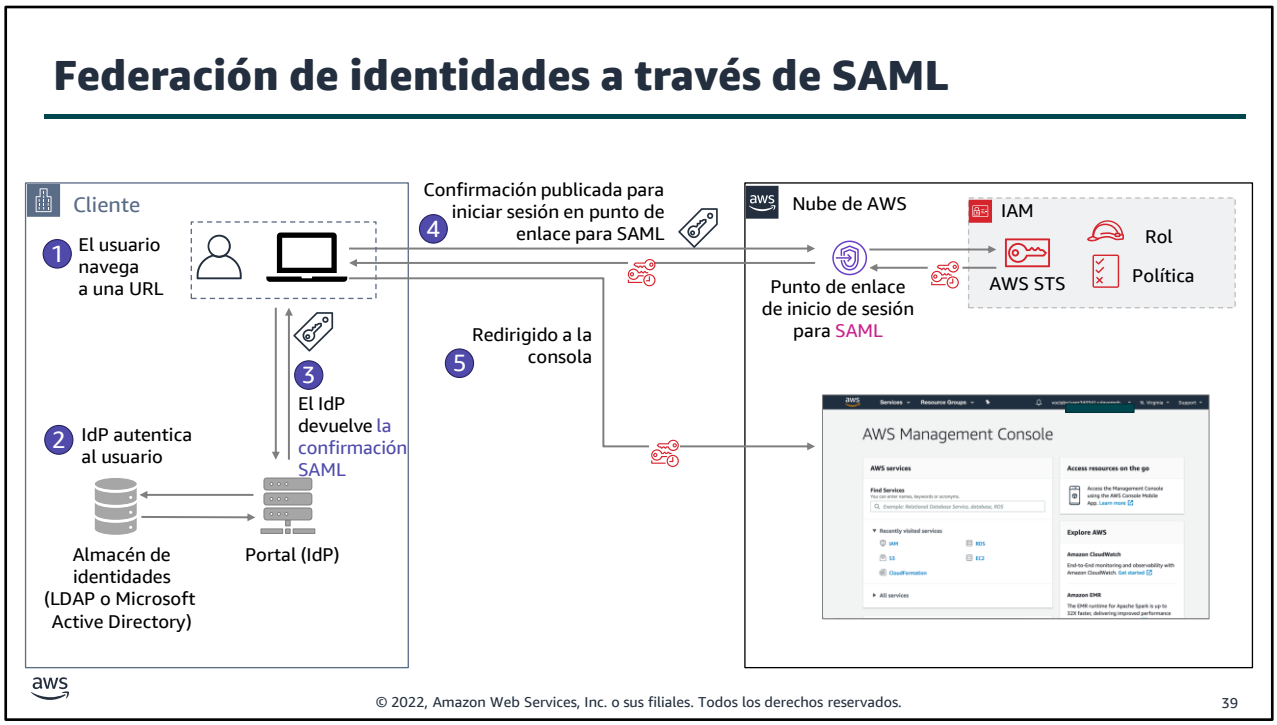
La federación de identidades se puede lograr de tres maneras. La primera forma es utilizar un IdP corporativo (como Microsoft Active Directory) o una aplicación de agente de identidades personalizada. Cada opción utiliza AWS STS. El segundo enfoque es crear una integración que utilice Security Assertion 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.



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.



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.

Amazon Cognito

- 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

aws

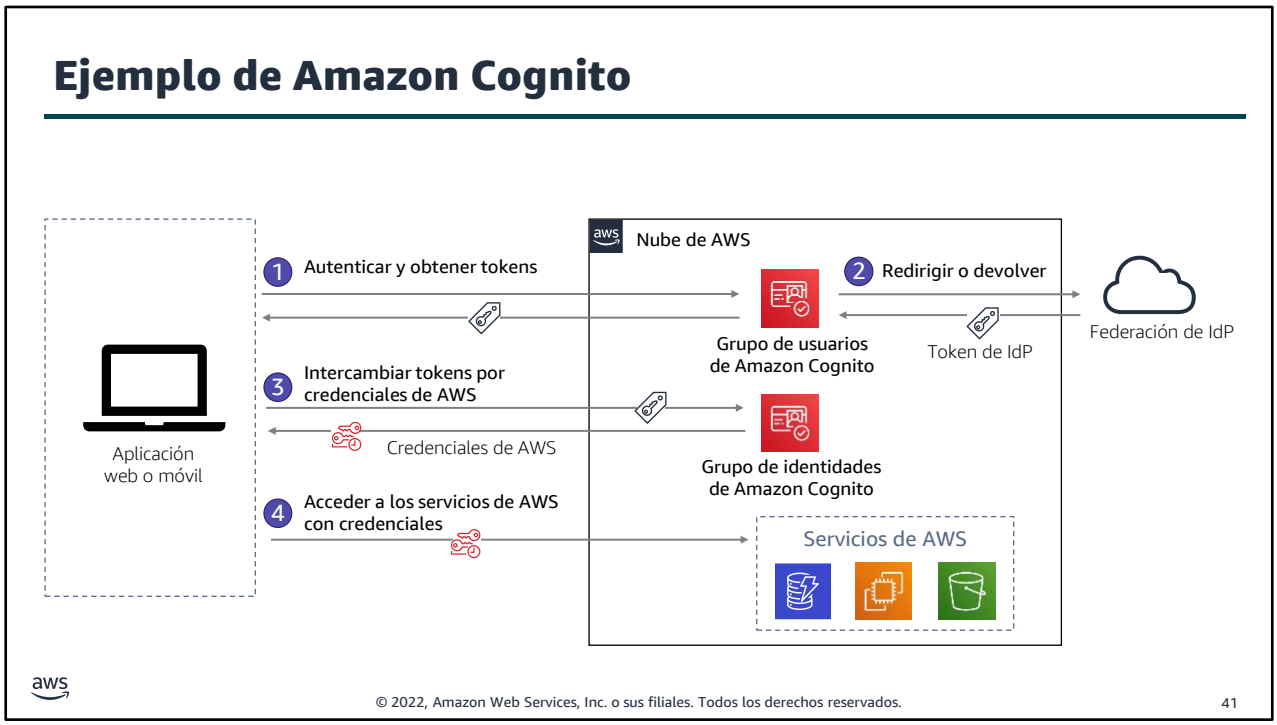
 40

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.

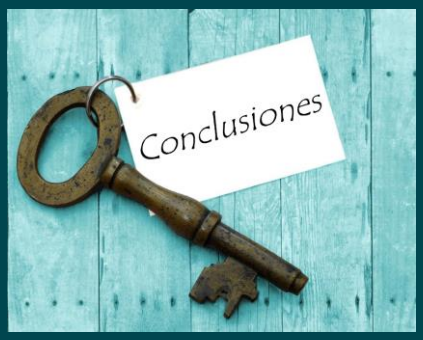
Losgrupos de identidades permiten la creación de identidades únicas y la asignación de permisos para los usuarios. Con un grupo de identidades, los usuarios pueden obtener credenciales temporales de AWS para acceder a los servicios o recursos de AWS. Los grupos de identidades pueden comunicarse con el 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).



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.

Conclusiones importantes de la Sección 4

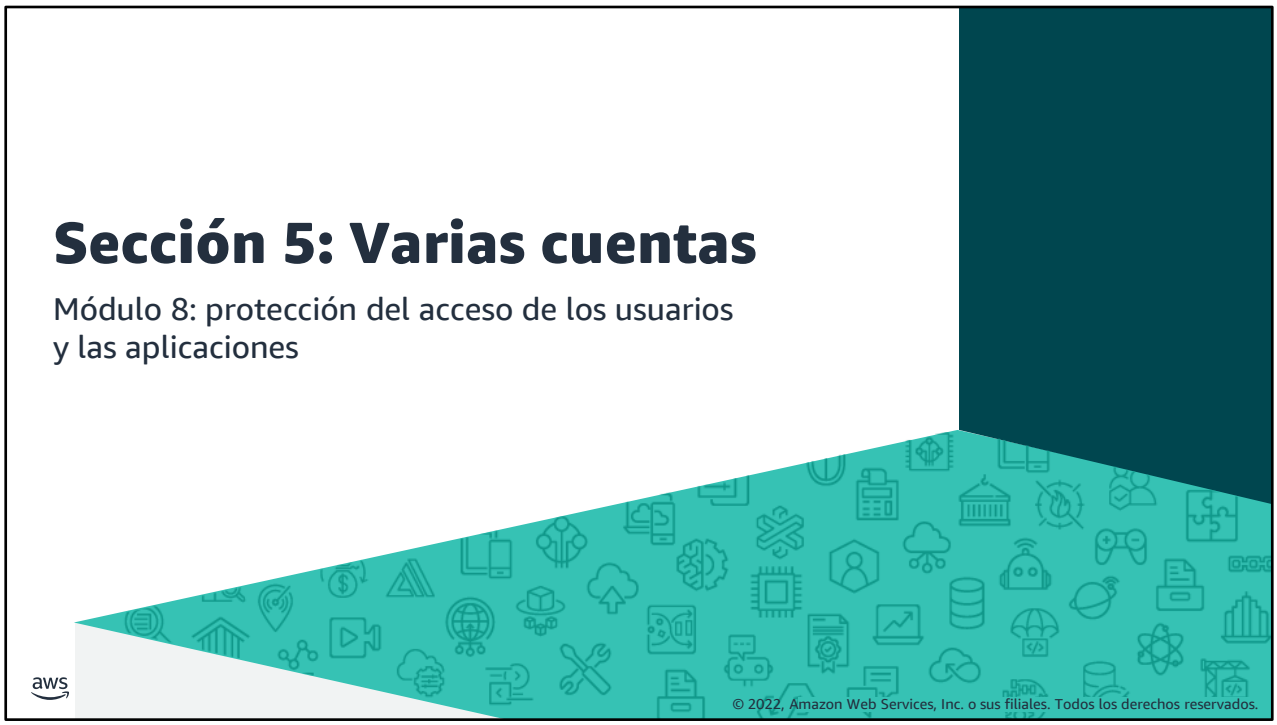


- Los roles de IAM proporcionan credenciales de seguridad temporales que puede asumir una persona, aplicación o servicio
- El AWS Security Token Service (AWS STS) le permite solicitar credenciales temporales de AWS
- Con la federación de identidades, la autenticación del usuario es externa a la cuenta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

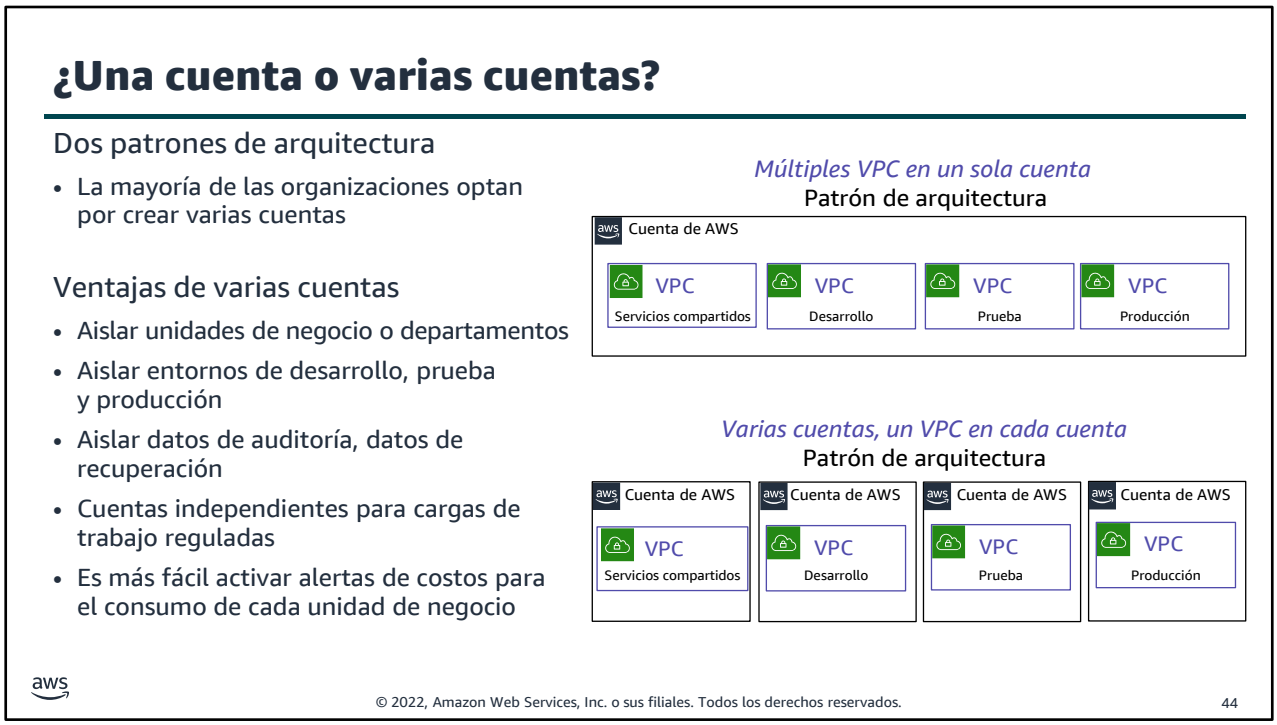
 42

Entre los aprendizajes clave de esta sección del módulo, 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.



Introducción a la Sección 5: varias cuentas



Cuando utiliza AWS para brindar soporte a los diferentes equipos y departamentos de una organización, puede elegir entre dos patrones de arquitectura generales para aislar y separar los recursos que utiliza cada equipo.

El primer patrón consiste en definir varias nubes privadas virtuales (VPC) en una cuenta única de AWS. Si prefiere una administración de seguridad de la información centralizada con una sobrecarga mínima, puede optar por utilizar una cuenta única de AWS.

El segundo patrón consiste en crear varias cuentas de AWS y definir una VPC en cada una de ellas. En la práctica, las organizaciones grandes y pequeñas tienden a crear varias cuentas para sus organizaciones. Por ejemplo, podrían crear cuentas individuales para varias unidades de negocio. También podrían crear cuentas independientes para sus recursos de desarrollo, prueba y producción.

Cuando los clientes utilizan cuentas de AWS separadas (generalmente con facturación unificada) para los recursos de desarrollo y producción, les permite separar claramente diferentes tipos de recursos. También puede proporcionar algunos beneficios de seguridad.

Alternativamente, si su empresa mantiene entornos separados para producción, desarrollo y pruebas, puede configurar tres cuentas de AWS y tener una cuenta para cada entorno. Además, si tiene varios departamentos autónomos, también puede crear cuentas de AWS independientes para cada parte autónoma de la organización.

Cuando utiliza varias cuentas, una estrategia más eficiente es crear una cuenta de AWS única para los recursos comunes del proyecto. Los recursos comunes pueden incluir servicios del Sistema de nombres de dominio (DNS), Microsoft Active Directory y sistemas de administración de

contenidos (CMS). También podría separar cuentas para los proyectos o departamentos autónomos. Esta estrategia le permite asignar permisos y políticas para cada departamento o cuenta de proyecto, y otorgar acceso a recursos entre cuentas.

Desafíos para administrar varias cuentas

- Administración de seguridad entre cuentas
  - Replicación de políticas de IAM
- Creación de cuentas nuevas
  - Implica muchos procesos manuales
- Consolidación de facturación
- Se necesita gobernanza centralizada para garantizar la coherencia

45

Aunque la mayoría de las organizaciones optan por utilizar varias cuentas de AWS, esa elección conlleva algunos desafíos.

Primero, debe determinar cómo administrar eficazmente la seguridad en todas sus cuentas. Si replica las políticas de IAM que definió en todas las cuentas para garantizar la coherencia, podría implicar automatización personalizada, esfuerzo manual o ambos.

Además, es posible que se le solicite constantemente que cree más cuentas. Se necesita tiempo para crear estas cuentas manualmente. También puede resultar difícil realizar un seguimiento de todas las cuentas y el propósito de cada cuenta.

También puede ser un desafío determinar a qué centro de costos de la organización se le debe facturar, por qué recursos y en qué cuentas. Y, por último, es posible que también desee lograr la gobernanza centralizada que se necesita para garantizar la coherencia.

Administra varias cuentas con AWS Organizations

Administrar y aplicar políticas de manera centralizada entre varias cuentas de AWS.

- Administración de cuentas basada en grupos
- Acceso a los servicios de AWS basado en políticas
- Administración y creación automatizada de cuentas
- Facturación unificada
- Basado en API

 46

AWS ofrece un servicio que está diseñado para abordar estos desafíos de administración.

AWS Organizations es un servicio administrado para la administración de cuentas. Una organización es una entidad que se crea para integrar, ver de forma centralizada y administrar todas las cuentas de AWS. Usted determina la funcionalidad de una organización a través del conjunto de funciones que habilita.

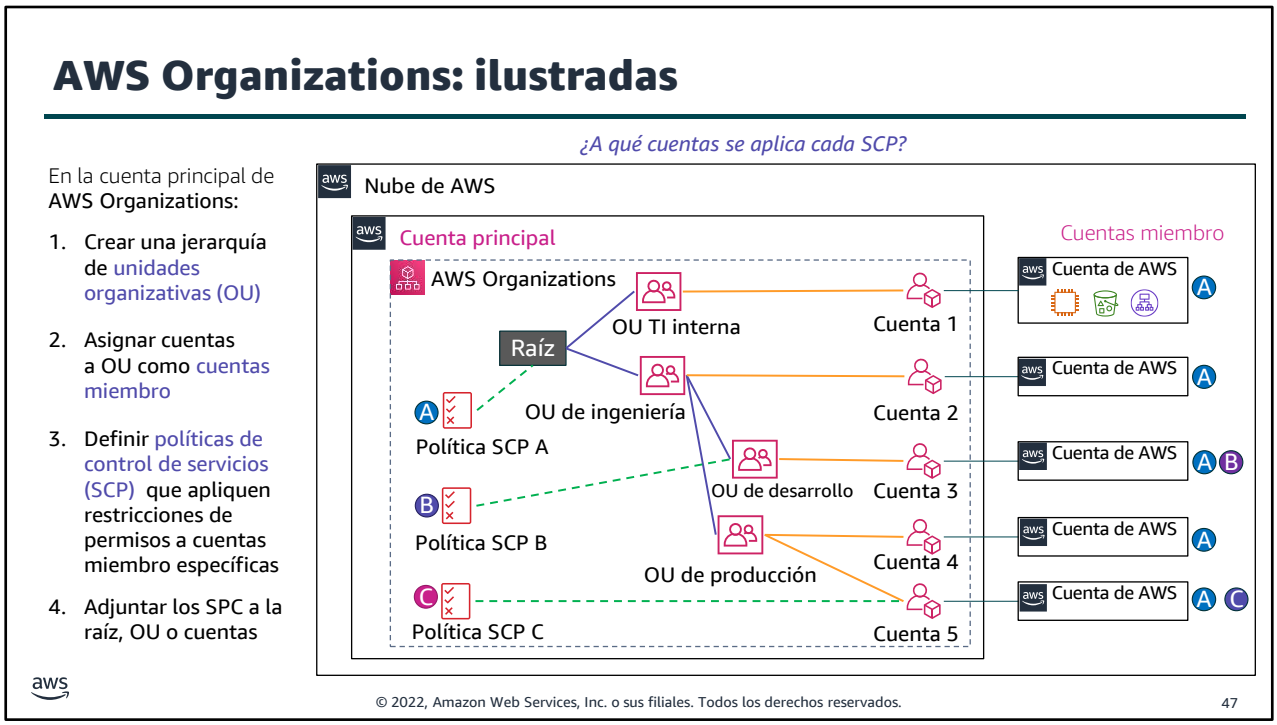
Organizations lo ayuda a administrar políticas para varias cuentas de AWS. Puede utilizar el servicio para crear grupos de cuentas. Para asegurarse de que se aplican las políticas correctas en toda la cuenta, adjunte políticas a un grupo.

Puede crear grupos de cuentas AWS y luego aplicar diferentes políticas a cada grupo.

Las API de Organizations pueden crear nuevas cuentas mediante programación y agregarlas a un grupo. Las políticas que se adjuntan al grupo se aplican de manera automática a la nueva cuenta.

También puede configurar un único método de pago para todas las cuentas de AWS de su empresa mediante la facturación unificada. Con la facturación unificada, puede ver una vista combinada de los cargos en los que incurren todas sus cuentas.

Finalmente, puede administrar el uso de los servicios de AWS a nivel de API. Por ejemplo, puede aplicar una política a un grupo de cuentas que solo permitirá a los usuarios de IAM de esas cuentas leer datos de buckets de S3.



Este es un ejemplo de AWS organization. Se define dentro de una cuenta de AWS normal a la que se hace referencia en la diapositiva como la cuenta principal porque en ella se define la organización de AWS.

Cuando crea una *organización*en la cuenta principal, la organización *crea automáticamente un contenedor principal que se denomina* raíz. Debajo de cada raíz de la organización, puede definir *unidades organizativas,* que también se conocen como *OU*. Cada OU es un contenedor de *cuentas miembro*. Una OU también puede contener otras OU y esas OU pueden contener más cuentas. Esta función le permite crear una jerarquía en forma de árbol. Puede pensar en la raíz y OU como ramas que se extienden y terminan en cuentas, que son como las hojas de un árbol.

Para configurarcontroles de acceso entre cuentas, debe entonces definir *políticas de control de servicios* (SCP). Adjunte cada política al lugar adecuado en la jerarquía de OU y cuentas. La política se aleja de la raíz y afecta a todas las OU y cuentas debajo de ella. Por lo tanto, si aplica una SCPa la raíz (como *la Política A de SCP en el ejemplo), se aplicará a todas las OU y cuentas de la organización.* Puede adjuntar la SCP a la raíz, a cualquier OU o una cuenta individual.

Recuerde que al igual que las políticas de IAM, las SCP solo otorgarán acceso si está permitido explícitamente y no está denegado explícitamente por cualquier otra SCP o política de IAM que se aplique al usuario. Por ejemplo, supongamos que la Política A de SCP, que se aplica a la raíz de la organización, establece más restricciones en un determinado servicio o conjunto de recursos que la Política C de SCP. Entonces, los usuarios en la Cuenta 5 están sujetos a los permisos más restrictivos establecidos por la Política A. Del mismo modo, si alguna política de IAM a nivel de cuenta individual deniega explícitamente cualquier acción para el usuario, estas políticas de IAM anulan cualquier permiso en las SCP que se otorgan a la cuenta.

Ejemplos de usos de SCP

- Características de las políticas de control de servicios (SCP)
  - Le permiten controlar a qué servicios pueden acceder los usuarios de IAM en cuentas miembro
  - El administrador local no puede anular los SCP
  - Las políticas de IAM definidas en cuentas individuales aún se aplican

- Ejemplos de usos de SCP
  - Crear una política que *bloquee* el acceso al servicio o acciones específicas
    - Ejemplo: impedir que los usuarios deshabiliten AWS CloudTrail en todas las cuentas miembro
  - Crear una política que *permita* el acceso completo a servicios específicos
    - Ejemplo: permitir el acceso completo a Amazon EC2 y CloudWatch
  - Crear una política que *imponga el etiquetado* de los recursos

aws

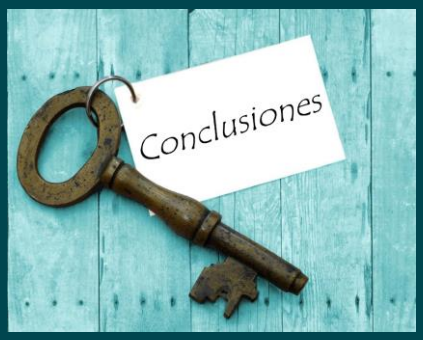
48

Las políticas de control de servicios (SCP) le permiten controlar a qué servicios pueden acceder los usuarios de IAM en cuentas miembro. Supongamos que tiene políticas específicas que desea aplicar en varias cuentas. Es más fácil definir estas políticas en una SCP que replicar estas configuraciones de permisos en documentos de políticas de IAM en cada cuenta.

Se deben usar las SCP con políticas de IAM que se definen en cada cuenta individual. Puede pensar que las SCP proporcionan límites generales en torno a los servicios y permisos generales a los que se debe permitir o denegar el acceso a los usuarios. Luego, puede utilizar las políticas de IAM para establecer controles de acceso más detallados que son específicos de cuentas individuales.

Puede crear SCP que bloqueen (o denieguen) el acceso a determinados servicios. También puede definir SCP que permitan el acceso a determinados servicios. Finalmente, puede decidir crear un SCP que imponga el etiquetado de recursos. Al hacerlo, su estrategia de etiquetado para control de acceso o asignación de costos puede seguir siendo efectiva cuando se crean nuevos recursos en sus cuentas.

Conclusiones importantes de la Sección 5



- Puede utilizar varias cuenta de AWS para aislar unidades de negocio, entornos de desarrollo y pruebas, cargas de trabajo reguladas y datos de auditoría
- AWS Organizations le permite configurar la creación automatizada y la facturación unificada
- Puede configurar controles de acceso entre cuentas a través de políticas de control de servicios (SCP)

 49

Entre los aprendizajes clave de esta sección del módulo, se incluyen los siguientes:

- Puede utilizar varias cuentas de AWS para aislar unidades de negocio, entornos de desarrollo y pruebas, cargas de trabajo reguladas y datos de auditoría
- AWS Organizations le permite configurar la creación automatizada de cuentas y la facturación unificada
- Puede configurar controles de acceso entre cuentas a través de políticas de control de servicios (SCP)

Módulo 8. Laboratorio de desafíos: control del acceso a las cuentas de AWS mediante IAM

 50

Ahora completará el Módulo 8: laboratorio de desafíos: control del acceso a las cuentas de AWS mediante IAM.

La necesidad empresarial: control de acceso de usuarios

La cafetería debe definir qué nivel de acceso deben tener los usuarios entre los recursos de la nube. Luego, deben implementar estos controles de acceso en la cuenta de AWS.

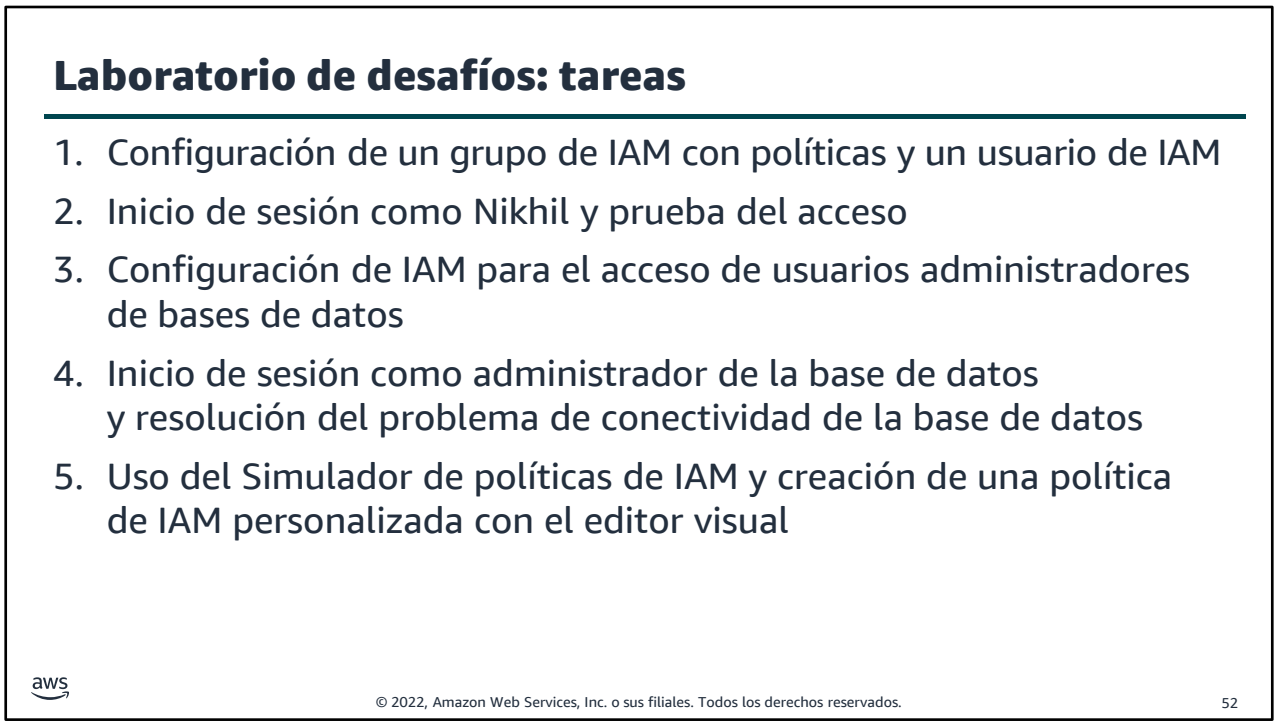
Cuando Mateo visitó la cafetería recientemente, le contó a **Sofía acerca de las funciones del servicio de IAM.** Ella planifica utilizar IAM para lograr su objetivo.

aws

 51

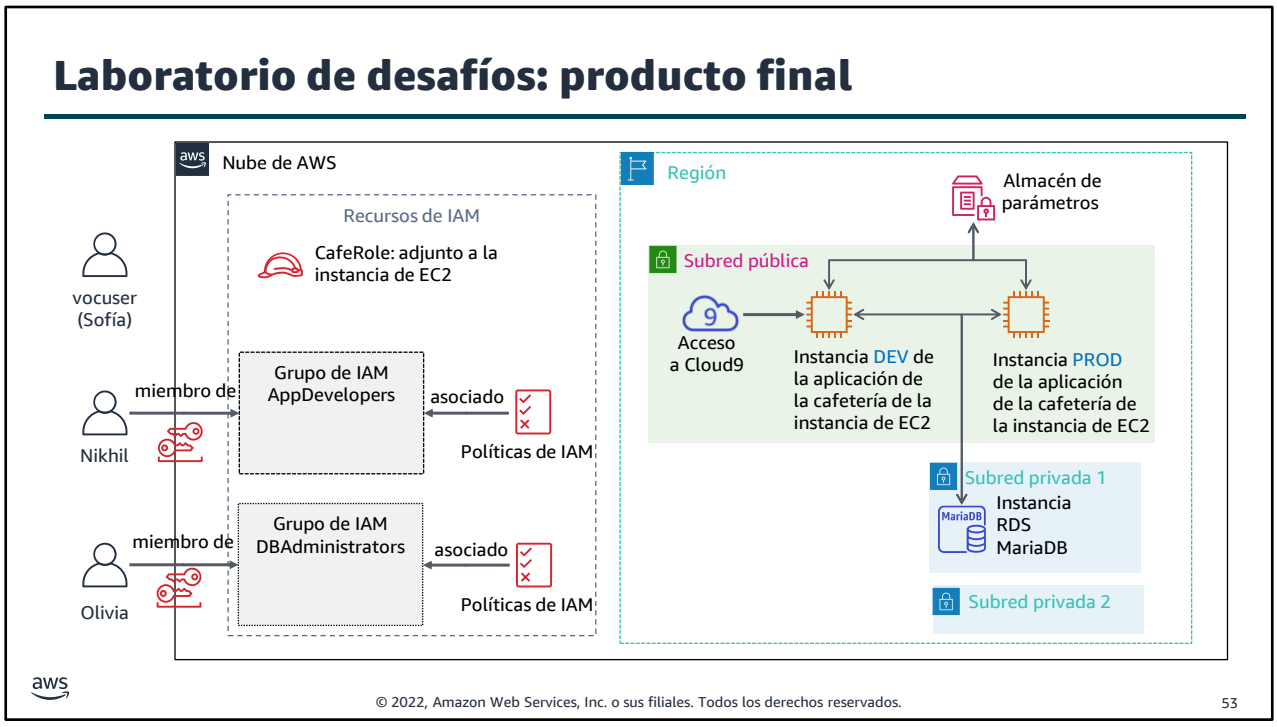
Después de hablar con Mateo sobre la infraestructura de AWS de la cafetería, Sofía se dio cuenta de que debía abordar algunos problemas básicos de seguridad sobre la forma en que el personal de la cafetería ha estado utilizando la cuenta de AWS.

La cafetería es ahora lo suficientemente grande como para que los miembros del equipo que crean, mantienen o acceden a aplicaciones en AWS se especialicen en roles (como desarrollador o administrador de bases de datos). Hasta ahora, no se habían esforzado por definir claramente qué nivel de acceso debería tener cada usuario en función de sus roles y responsabilidades.



En este laboratorio de desafíos, realizará las siguientes tareas:

1. Configuración de un grupo de IAM con políticas y un usuario de IAM
2. Inicio de sesión como Nikhil y prueba del acceso
3. Configuración de IAM para el acceso de usuarios administradores de bases de datos
4. Inicio de sesión como administrador de la base de datos y resolución del problema de conectividad de la base de datos
5. Uso del Simulador de políticas de IAM y creación de una política de IAM personalizada con el editor visual



El diagrama resume lo que construirá después de completar el laboratorio.

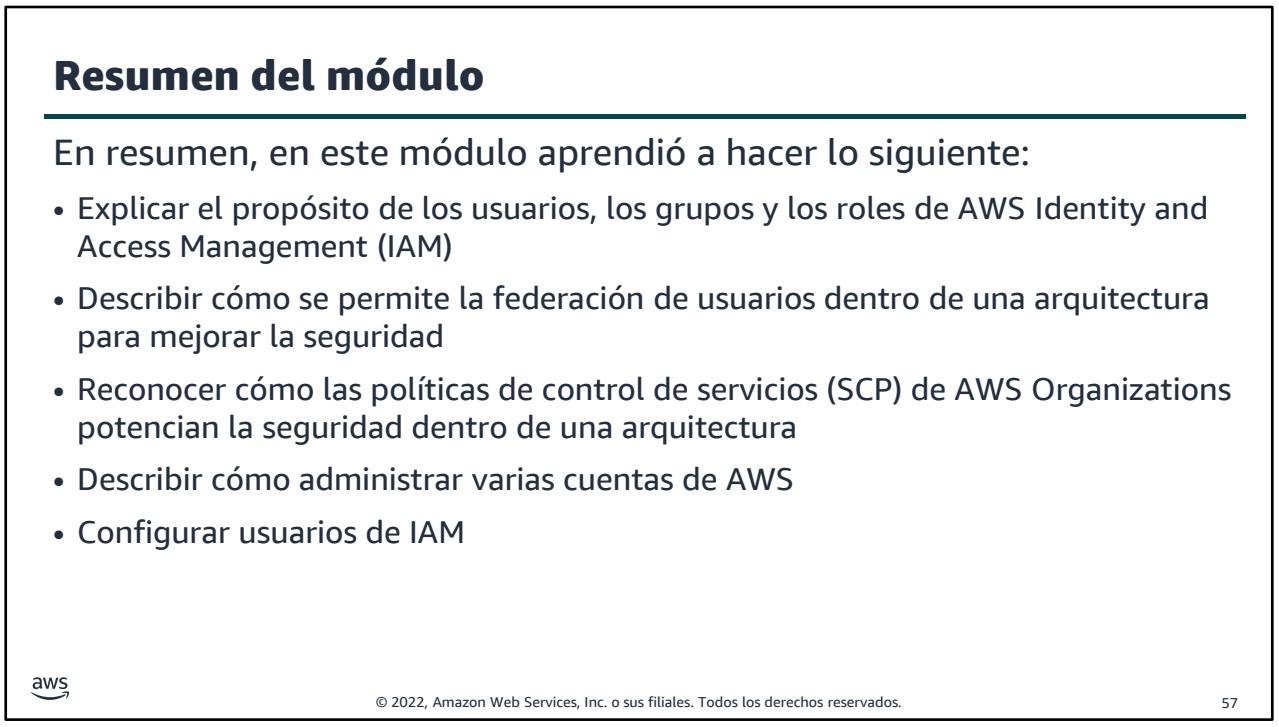
Es momento de comenzar con el laboratorio de desafíos.

Análisis posterior del laboratorio de desafíos: Aprendizajes clave

 55

El instructor ahora puede optar por dirigir una conversación sobre los aprendizajes clave de este laboratorio de desafíos después de que lo haya completado.

Ahora es el momento de revisar el módulo y concluir con una evaluación de conocimientos y una discusión sobre una pregunta del examen de certificación de práctica.



En resumen, en este módulo aprendió a hacer lo siguiente:

- Explicar el propósito de los usuarios, los grupos y los roles de AWS Identity and Access Management (IAM).
- Describir cómo se permite la federación de usuarios dentro de una arquitectura para mejorar la seguridad
- Reconocer cómo las políticas de control de servicios (SCP) de AWS Organizations potencian la seguridad dentro de una arquitectura
- Describir cómo administrar varias cuentas de AWS
- Configurar usuarios de IAM

Completar la evaluación de conocimientos

Ahora es el momento de completar la evaluación de conocimientos para este módulo.

Pregunta de examen de ejemplo

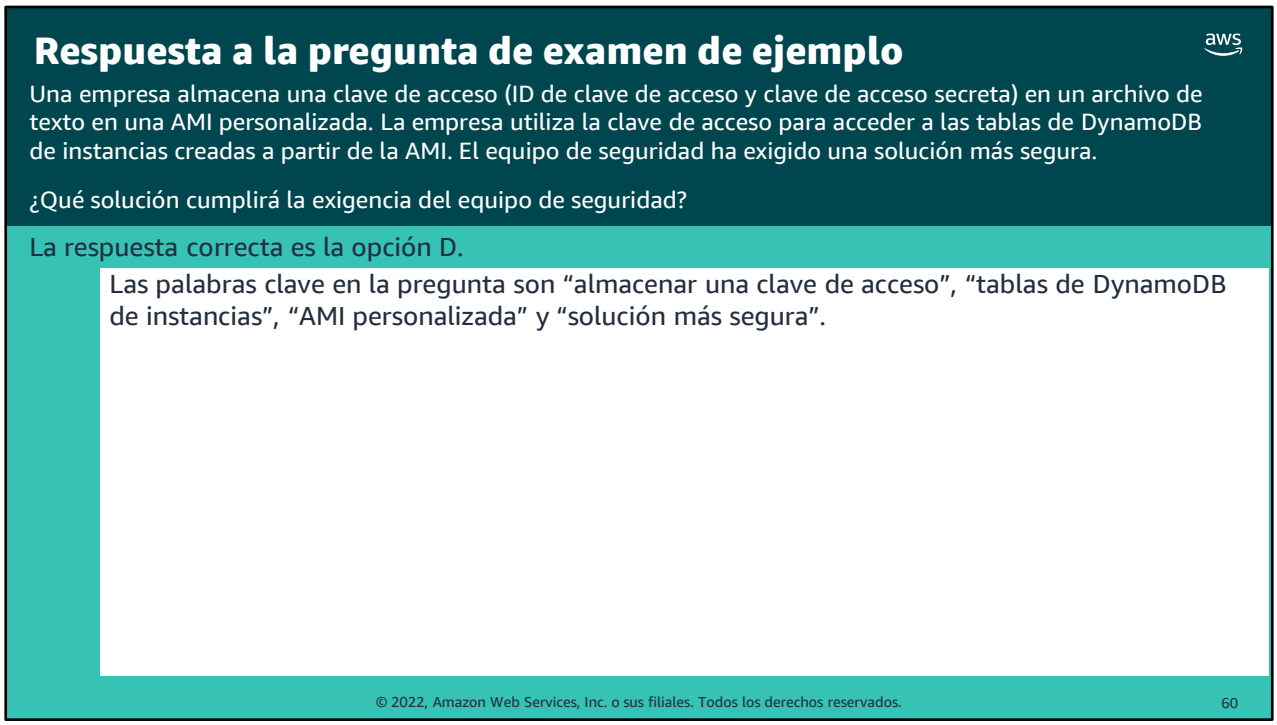
Una empresa almacena una clave de acceso (ID de clave de acceso y clave de acceso secreta) en un archivo de texto en una AMI personalizada. La empresa utiliza la clave de acceso para acceder a las tablas de DynamoDB de instancias creadas a partir de la AMI. El equipo de seguridad ha exigido una solución más segura.

¿Qué solución cumplirá la exigencia del equipo de seguridad?

Opción	Respuesta
A	Colocar la clave de acceso en un bucket S3 y recuperarla al arrancar desde la instancia.
B	Pasar la clave de acceso a las instancias a través de los datos del usuario de la instancia.
C	Obtener la clave de acceso de un servidor de claves iniciado en una subred privada.
D	Crear un rol de IAM con permisos para acceder a la tabla e iniciar todas las instancias con el nuevo rol.

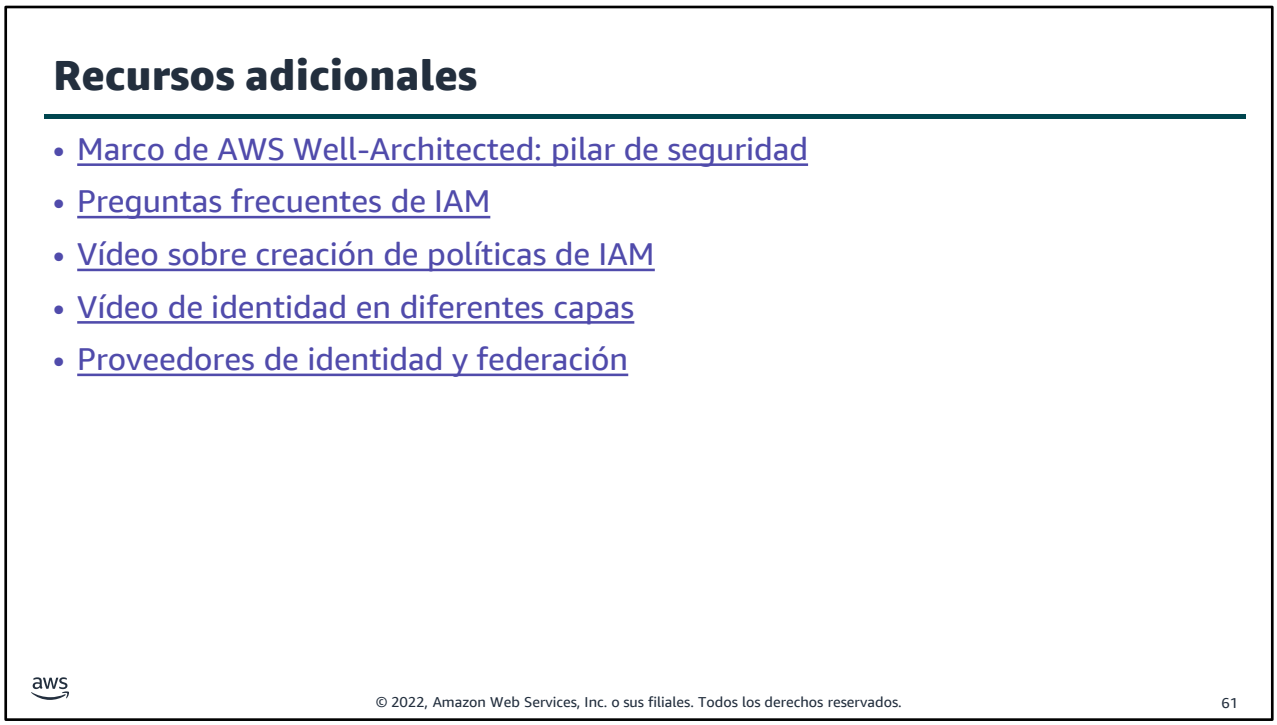
 59

Mire las opciones de respuesta y descártelas según las palabras clave.



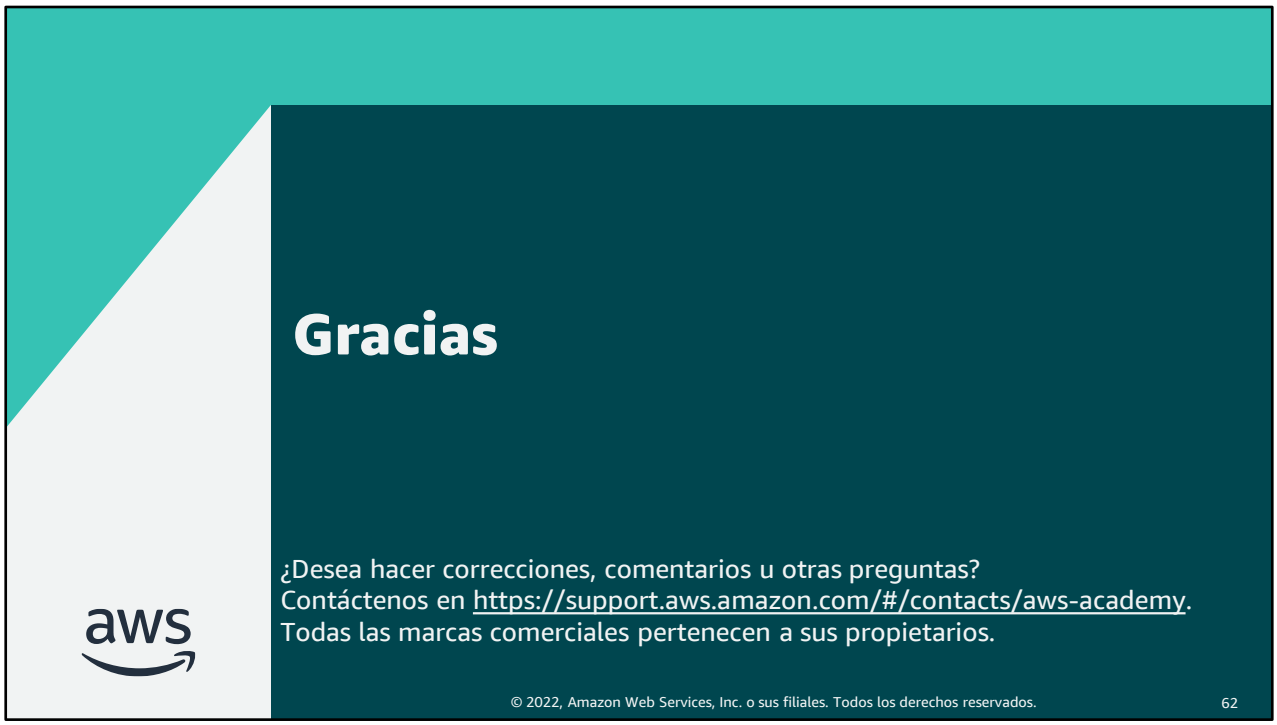
Las siguientes son las palabras clave que debe reconocer **: “almacenar una clave de acceso”, “tablas DynamoDB de instancias”, “AMI personalizada” y “solución más segura”.**

La respuesta correcta es D. Los roles de IAM para instancias EC2 permiten que las aplicaciones que se ejecutan en la instancia accedan a recursos AWS sin la necesidad de crear y almacenar claves de acceso. Cualquier solución que implique la creación de una clave de acceso introduce la complejidad de administrar ese secreto.



Si desea obtener más información sobre los temas tratados en este módulo, es posible que le resulten útiles los siguientes recursos adicionales:

- Marco de AWS Well-Architected: pilar de seguridad
- Preguntas frecuentes de IAM
- Vídeo sobre creación de políticas de IAM
- Vídeo de identidad en diferentes capas
- Proveedores de identidad y federación



Gracias por completar este módulo.

