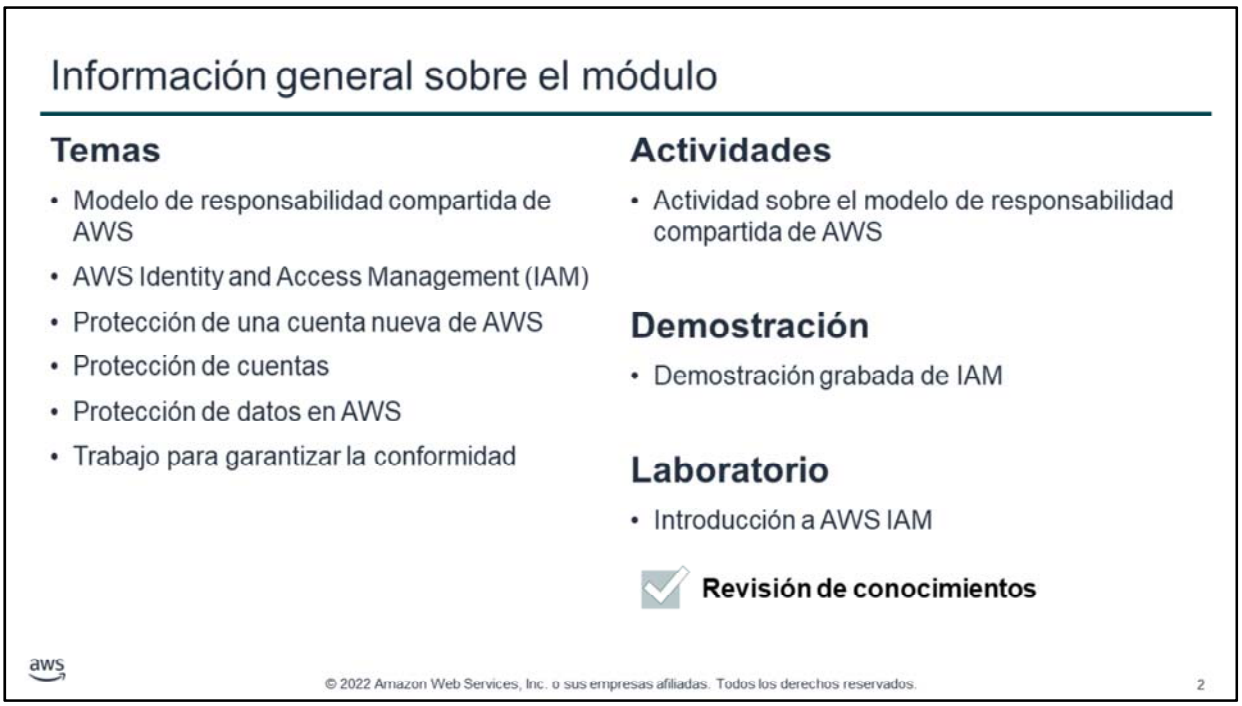
Bienvenido al Módulo 4: Seguridad en la nube de AWS.

La seguridad es la máxima prioridad en Amazon Web Services (AWS). AWS ofrece un entorno de informática en la nube escalable diseñado para proporcionar alta disponibilidad y fiabilidad con herramientas que le permiten ejecutar una amplia variedad de aplicaciones. Además de mantener la confianza de sus clientes, para AWS también es fundamental ayudar a proteger la confidencialidad, integridad y disponibilidad de sus sistemas y datos. Este módulo ofrece una introducción al enfoque de seguridad de AWS, que incluye tanto los controles en el entorno de AWS como algunos de los productos y las características de AWS que los clientes pueden utilizar para cumplir sus objetivos de seguridad.



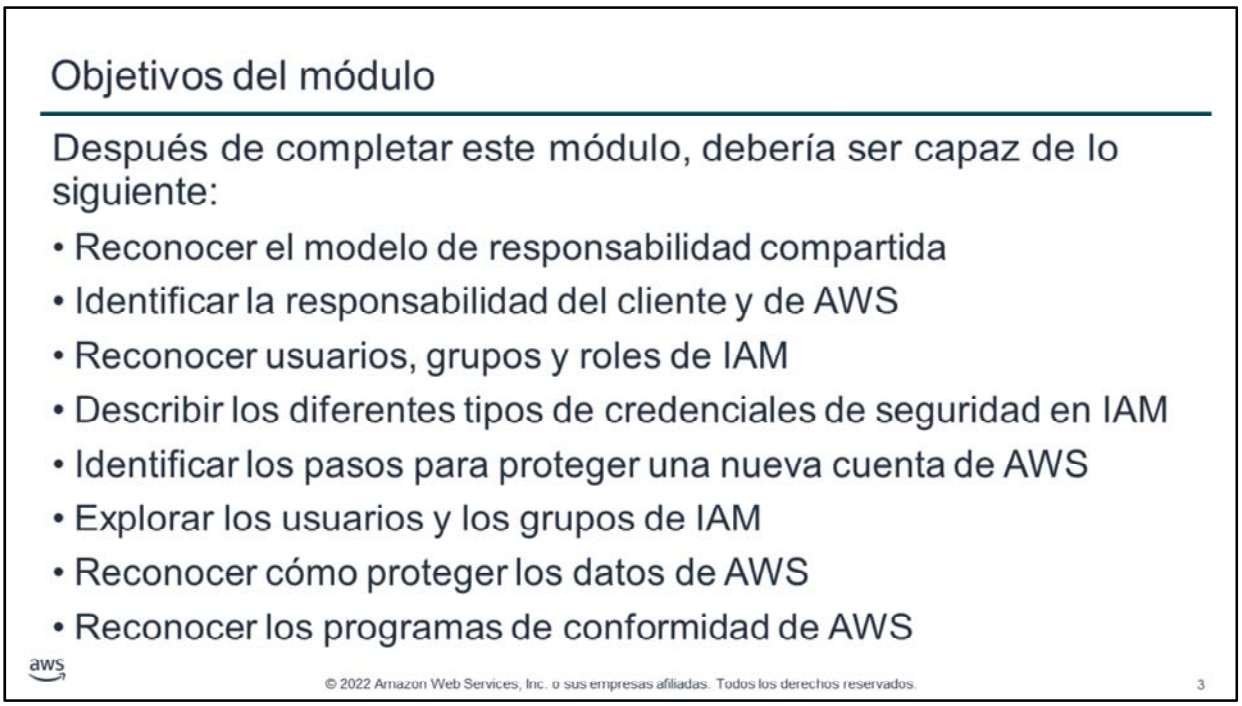
En este módulo, se abordarán los siguientes temas:

- Modelo de responsabilidad compartida de AWS
- AWS Identity and Access Management (IAM)
- Protección de una cuenta nueva de AWS
- Protección de cuentas
- Protección de datos en AWS
- Trabajo para garantizar la conformidad
- Servicios y recursos de seguridad adicionales

La sección uno incluye una **actividad** impartida por un profesor sobre el modelo de responsabilidad compartida de AWS.

En la sección dos, se incluye una **demostración de IAM** grabada y, al final de esta misma sección, se incluye un **laboratorio práctico** que le ofrece experiencia en la configuración deIAM mediante el uso de la consola de administración de AWS.

Por último, se le pedirá que complete una **revisión de conocimientos** para comprobar su comprensión de los conceptos clave que se tratan en este módulo.



Después de completar este módulo, debería ser capaz de lo siguiente:

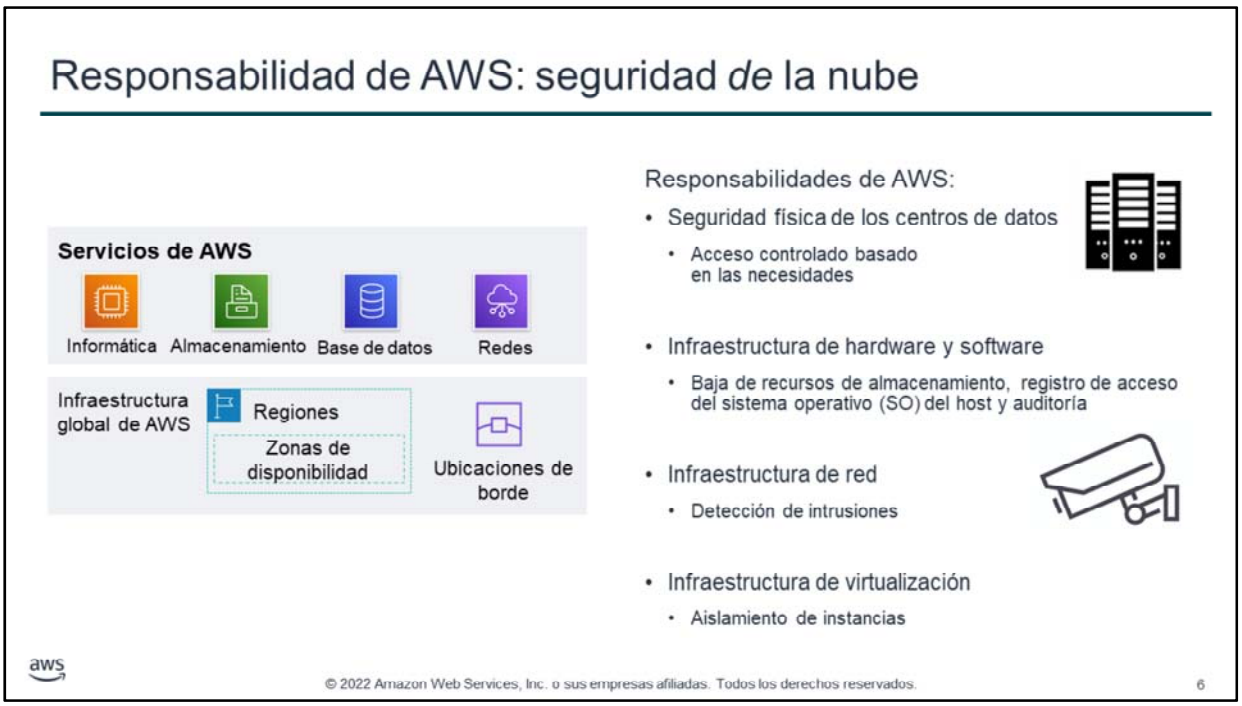
- Reconocer el modelo de responsabilidad compartida
- Identificar la responsabilidad del cliente y de AWS
- Reconocer usuarios, grupos y roles de IAM
- Describir los diferentes tipos de credenciales de seguridad en IAM
- Identificar los pasos para proteger una nueva cuenta de AWS
- Explorar los usuarios y los grupos de IAM
- Reconocer cómo proteger los datos de AWS
- Reconocer los programas de conformidad de AWS

Presentación de la sección 1: Modelo de responsabilidad compartida de AWS

La seguridad y la conformidad son una responsabilidad compartida entre AWS y el cliente. Este modelo de responsabilidad compartida está diseñado para ayudar a aliviar la carga operativa del cliente. Al mismo tiempo, para proporcionar la flexibilidad y el control del cliente que permite la implementación de soluciones de clientes en AWS, el cliente sigue siendo responsable de algunos aspectos de la seguridad general. La diferencia de quién es responsable entre lo que comúnmente se denomina *seguridad "de" la nube* frente a *seguridad "en" la nube*.

AWS opera, administra y controla los componentes desde la capa de virtualización de software hasta la seguridad física de las instalaciones en las que operan los servicios de AWS. **AWS es responsable** de proteger la infraestructura en la que se ejecutan todos los servicios ofrecidos en la nube de AWS. Esta infraestructura está conformada por el hardware, el software, las redes y las instalaciones que ejecutan los servicios de la nube de AWS.

El cliente es responsable del cifrado de los datos en reposo y los datos en tránsito. El cliente también debe asegurarse de que la red esté configurada para la seguridad y de que las credenciales de seguridad y los inicios de sesión se administren de forma segura. Además, el cliente es responsable de la configuración de los grupos de seguridad y de la configuración del sistema operativo que se ejecuta en las instancias informáticas que lanzan (incluidas las actualizaciones y los parches de seguridad).



AWS es responsable de la seguridad ***de*** la nube. ¿Qué significa esto?

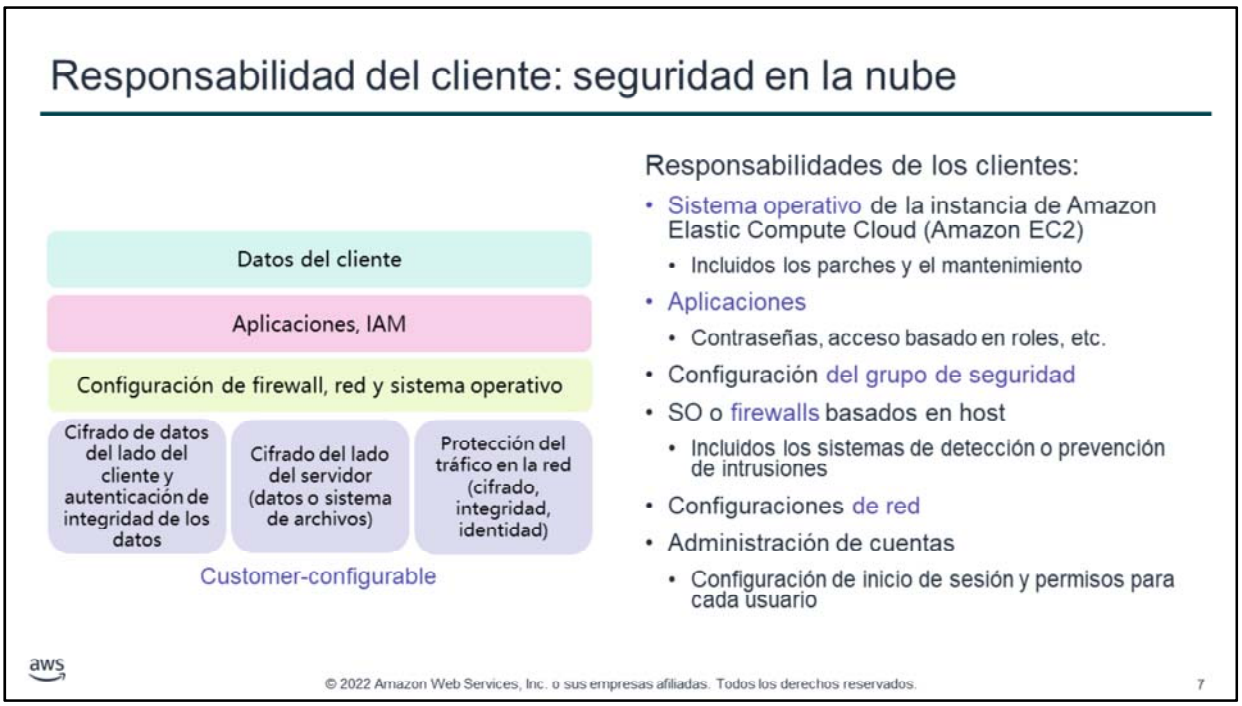
En el modelo de responsabilidad compartida, AWS opera, administra y controla los componentes desde el sistema operativo de alojamiento bare metal y la capa de virtualización del hipervisor hasta la seguridad física de las instalaciones en las que operan los servicios. Esto significa que AWS se encarga de proteger la infraestructura global que ejecuta todos los servicios ofrecidos en la nube de AWS. La infraestructura global incluye regiones de AWS, zonas de disponibilidad y ubicaciones de borde.

AWS es responsable de la infraestructura física que aloja sus recursos, que incluye lo siguiente:

- **Seguridad física de los centros de datos** con acceso controlado basado en las necesidades en instalaciones sin identificación, con guardias de seguridad permanentes, autenticación de dos factores, revisión y registro de accesos, videovigilancia, y destrucción y desmagnetización de discos.
- **Infraestructura de hardware**, como servidores, dispositivos de almacenamiento y otros dispositivos de los que dependen los servicios de AWS.
- **Infraestructura de software**, que aloja sistemas operativos, aplicaciones de servicios y software de virtualización.
- **Infraestructura de red**, como routers, conmutadores, balanceadores de carga,

firewalls y cables. AWS también monitorea la red en límites externos, protege los puntos de acceso y proporciona infraestructura redundante con detección de intrusiones de forma constante.

La protección de esta infraestructura es la principal prioridad de AWS. Si bien no puede visitar los centros de datos o las oficinas de AWS para ver esta protección de primera mano, Amazon proporciona numerosos informes de auditores externos que han verificado nuestra conformidad con una variedad de normas y estándares de seguridad informática.



Mientras que AWS protege y mantiene la infraestructura de la nube, los clientes son responsables de la seguridad de todo lo que ponen ***en*** la nube.

El **cliente es responsable** de lo que se implementa a través del uso de los servicios de AWS y de las aplicaciones que están conectadas a AWS. Los pasos de seguridad que debe tomar dependerán de los servicios que utilice y de la complejidad del sistema.

Las responsabilidades de los clientes incluyen la selección y protección de cualquier sistema operativo de instancias, la protección de las aplicaciones que se lanzan en los recursos de AWS, las configuraciones de grupos de seguridad, las configuraciones de firewall, las configuraciones de red y la administración segura de cuentas.

Cuando los clientes utilizan los servicios de AWS, mantienen un control absoluto sobre su contenido. Los clientes son responsables de administrar los requisitos de seguridad de contenido críticos, entre los que se incluyen los siguientes:

- El contenido que eligen almacenar en AWS
- Los servicios de AWS que se utilizan con el contenido
- En qué país se almacena ese contenido
- El formato y la estructura de ese contenido y si está enmascarado, cifrado o es anónimo
- Quién tiene acceso a ese contenido y cómo se conceden, administran y revocan esos derechos de acceso

Los usuarios retienen el control de la seguridad que deciden implementar para proteger sus propios datos, entorno, aplicaciones, configuraciones de IAM y sistemas operativos.

La infraestructura como servicio (IaaS) se refiere a los servicios que proporcionan bloques de creación básicos para la TI en la nube, que incluyen, por lo general, acceso para configurar redes, equipos (virtuales o en hardware dedicado) y espacio de almacenamiento de datos. Los servicios en la nube que se pueden caracterizar como IaaS **proporcionan al cliente el mayor nivel de flexibilidad y control de administración** sobre los recursos de TI. Los servicios de IaaS son más similares a los recursos informáticos existentes en las instalaciones con los que muchos departamentos de TI están familiarizados actualmente.

Los servicios de AWS, como **Amazon EC2**, pueden clasificarse como **IaaS** y, por lo tanto, **exigir que el cliente realice todas las tareas de configuración y administración de seguridad necesarias**. Los clientes que implementan instancias EC2 son responsables de administrar el sistema operativo invitado (incluidas las actualizaciones y los parches de seguridad), cualquier software de aplicación instalado en las instancias y la configuración de los grupos de seguridad proporcionados por AWS.

Plataforma como servicio (PaaS) se refiere a los servicios que eliminan la necesidad de que el cliente administre la infraestructura subyacente (hardware, sistemas operativos, etc.). Los servicios PaaS permiten al cliente enfocarse completamente en la implementación y administración de aplicaciones. Los clientes no tienen que preocuparse por la adquisición de recursos, la planificación de capacidad, el mantenimiento del

software ni la implementación de parches.

Los servicios de AWS, como **AWS Lambda** y **Amazon RDS** pueden clasificarse como **PaaS** porque **AWS opera la capa de infraestructura, el sistema operativo y las plataformas**. Los clientes solo necesitan obtener acceso a los puntos de enlace para almacenar y recuperar datos. Con los servicios PaaS, los clientes son responsables de administrar sus datos, clasificar sus recursos y aplicar los permisos adecuados. Sin embargo, estos servicios actúan más como servicios administrados, ya que AWS se encarga de una mayor parte de los requisitos de seguridad. En lo que respecta a estos servicios, AWS gestiona las tareas de seguridad básicas, como la implementación de parches en la base de datos y el sistema operativo, la configuración del firewall y la recuperación de desastres.

Software como servicio (SaaS) se refiere a los servicios que proporcionan software alojado de forma centralizada que suele ser accesible a través de un navegador web, una aplicación móvil o una interfaz de programación de aplicaciones (API). El modelo de licencia para ofertas de SaaS suele ser de suscripción o pago por uso. Con las ofertas de SaaS, los clientes no necesitan administrar la infraestructura que respalda el servicio. Algunos servicios de AWS, como **AWS Trusted Advisor, AWS Shield** y **Amazon Chime**, podrían clasificarse como ofertas de SaaS, en función de sus características.

AWS Trusted Advisor es una herramienta en línea que analiza su entorno de AWS y proporciona orientación y recomendaciones en tiempo real para ayudarlo a aprovisionar sus recursos siguiendo las prácticas recomendadas de AWS. El servicio Trusted Advisor se ofrece como parte de su plan de AWS Support. Algunas de las características de Trusted Advisor son gratuitas para todas las cuentas, pero los clientes de Business Support y Enterprise Support tienen acceso al conjunto completo de comprobaciones y recomendaciones de Trusted Advisor.

AWS Shield es un servicio administrado de protección contra ataques de denegación de servicio distribuidos (DDoS) que protege las aplicaciones que se ejecutan en AWS. Ofrece detección permanente y mitigaciones directas automáticas que reducen el tiempo de inactividad y latencia de las aplicaciones, por lo que no hay necesidad de contar con AWS Support para disfrutar de la protección de DDoS. AWS Shield Advanced está

disponible para todos los clientes. Sin embargo, para ponerse en contacto con el equipo de respuesta de DDoS, los clientes deben contar con Enterprise Support o Business Support de AWS Support.

Amazon Chime es un servicio de comunicaciones que le permite hacer reuniones, conversar y realizar llamadas empresariales dentro y fuera de su organización, todo con una sola aplicación. Es un servicio de comunicaciones de pago por uso sin tarifas iniciales, compromisos ni contratos a largo plazo.

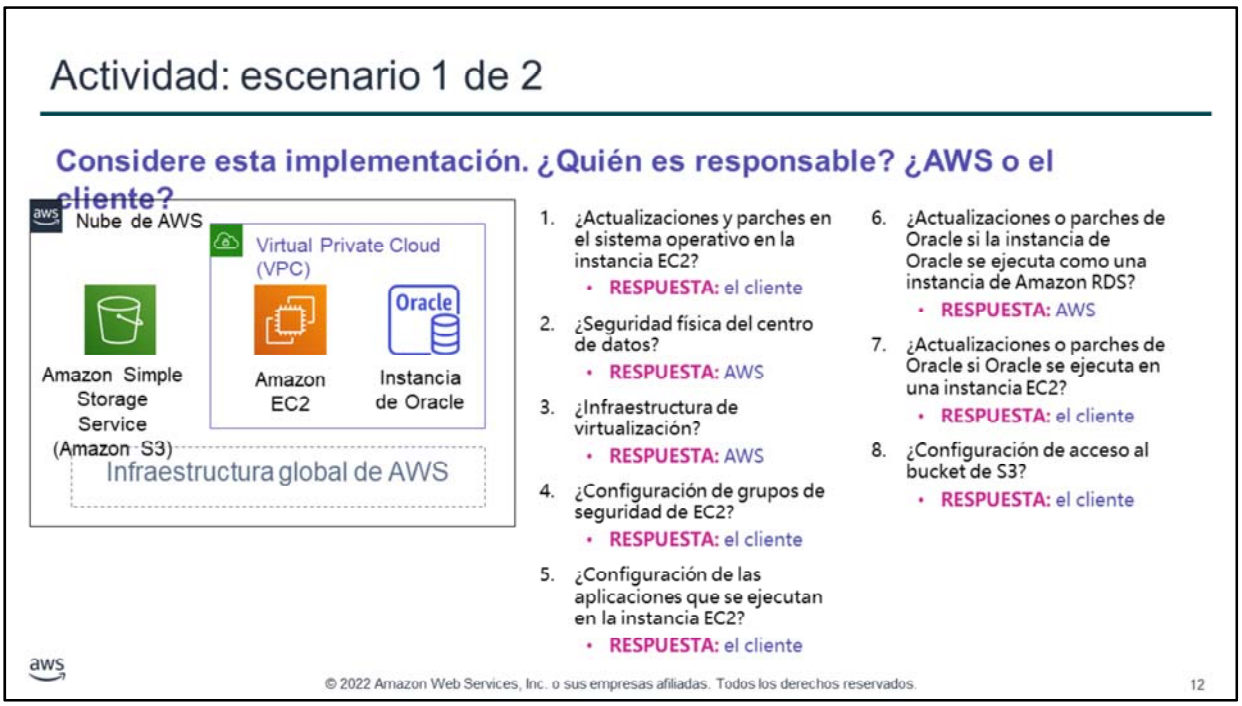
En esta actividad impartida por un profesor, se le presentarán dos situaciones. En cada situación, se le harán varias preguntas sobre quién es el responsable (AWS o el cliente) de garantizar la seguridad del elemento en cuestión. El profesor dirigirá la clase en un análisis de cada pregunta y revelará las respuestas correctas una por una.

Actividad: Escenario 1 de 2

Considere esta implementación. ¿Quién es responsable: AWS o el cliente?

1. ¿Actualizaciones y parches del sistema operativo en la instancia de EC2?
2. ¿Seguridad física de los centros de datos?
3. ¿Infraestructura de virtualización?
4. ¿Configuración del grupo de seguridad EC2?
5. ¿Configuración de aplicaciones que se ejecutan en la instancia de EC2?
6. ¿Actualizaciones o parches de Oracle si la instancia de Oracle se ejecuta como una instancia de Amazon RDS?
7. ¿Actualizaciones o parches de Oracle si Oracle se ejecuta en una instancia de EC2?
8. ¿Configuración de acceso al bucket de S3?

Considere el caso en el que un cliente utiliza los recursos y servicios de AWS que se muestran aquí. ¿Quién es responsable de mantener la seguridad? ¿AWS o el cliente?



Considere el caso en el que un cliente utiliza los servicios y recursos de AWS que se muestran aquí. ¿Quién es responsable de mantener la seguridad? ¿AWS o el cliente?

El cliente utiliza Amazon Simple Storage Service (Amazon S3) para almacenar datos. El cliente configuró una nube virtual privada (VPC) con Amazon Virtual Private Cloud (Amazon VPC). La instancia EC2 y la instancia de base de datos de Oracle que crearon se ejecutan en la VPC.

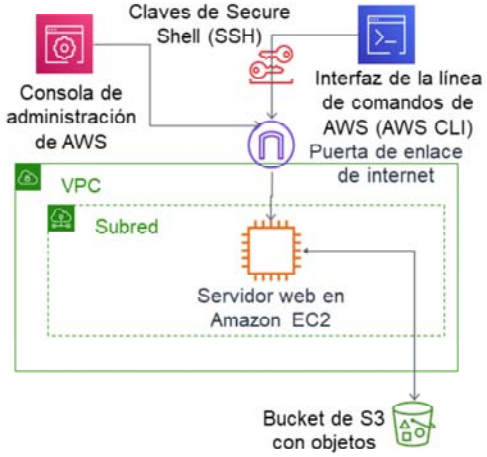
En este ejemplo, el cliente debe administrar el sistema operativo (SO) invitado que se ejecuta en la **instancia EC2**. Con el paso del tiempo, el sistema operativo invitado tendrá que actualizarse y aplicar parches de seguridad. Además, también se debe mantener todo software o utilidad de aplicación que el cliente haya instalado en la instancia de Amazon EC2. El cliente es responsable de configurar el firewall (o grupo de seguridad) de AWS que se aplica a la instancia de Amazon EC2. El cliente también es responsable de las configuraciones de **VPC** que especifican las condiciones de red en las que se ejecuta la instancia de Amazon EC2. Estas tareas son las mismas tareas de seguridad que realizaría el personal de TI, independientemente de dónde se encuentren sus servidores.

La instancia de Oracle de este ejemplo proporciona un caso práctico interesante en términos de responsabilidad de AWS o del cliente. **Si la base de datos se ejecuta en**

una instancia EC2, es responsabilidad del cliente aplicar parches y actualizaciones de software de Oracle. Sin embargo, **si la base de datos se ejecuta como una instancia de Amazon RDS**, AWS es el responsable de aplicar actualizaciones y parches de software de Oracle. Como Amazon RDS es un servicio de base de datos administrado, AWS se encarga de las tareas de administración de bases de datos que consumen mucho tiempo, como el aprovisionamiento, las copias de seguridad, los parches de software, el monitoreo y el escalado de hardware. Para obtener más información, consulte las prácticas recomendadas para ejecutar la base de datos de Oracle en AWS.

Actividad: Escenario 2 de 2

Considere esta implementación. ¿Quién es responsable: AWS o el cliente?

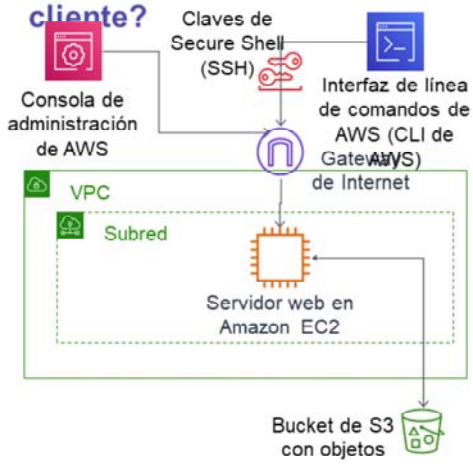


1. ¿Se asegura de que la Consola de administración de AWS no sea pirateada?
2. ¿Configurando la subred?
3. ¿Configurando la VPC?
4. ¿Protección contra las interrupciones de la red en las regiones de AWS?
5. Proteger las claves SSH
6. ¿Garantizar el aislamiento de la red entre los datos de los clientes de AWS?
7. ¿Garantizar una conexión de red de baja latencia entre el servidor web y el bucket de S3?
8. ¿Hacer cumplir la autenticación multifactor para todos los inicios de sesión de los usuarios?

Ahora, considere este caso adicional en el que un cliente utiliza los servicios y recursos de AWS que se muestran aquí. ¿Quién es responsable de mantener la seguridad? ¿AWS o el cliente?

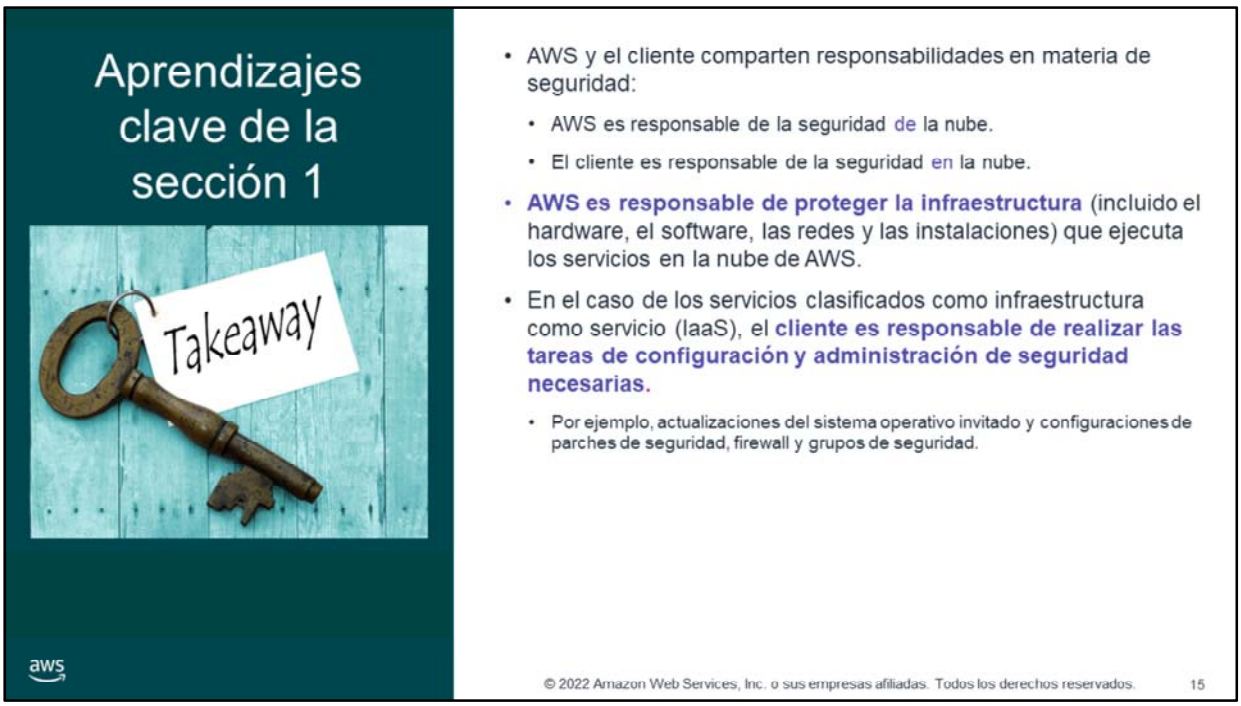
Actividad: escenario 2 de 2

Considere esta implementación. ¿Quién es responsable? ¿AWS o el cliente?



1. ¿Garantizar que la consola de administración de AWS no sea pirateada?
   - **RESPUESTA:** AWS
2. ¿Configurar la subred?
   - **RESPUESTA:** el cliente
3. ¿Configurar la VPC?
   - **RESPUESTA:** el cliente
4. ¿Proteger frente a interrupciones de red en las regiones de AWS?
   - **RESPUESTA:** AWS
5. ¿Proteger las claves SSH?
   - **RESPUESTA:** el cliente
6. ¿Garantizar el aislamiento de red entre los datos de los clientes de AWS?
   - **RESPUESTA:** AWS
7. ¿Garantizar una conexión de red de baja latencia entre el servidor web y el bucket de S3?
   - **RESPUESTA:** AWS
8. ¿Requerir la autenticación multifactor para todos los inicios de sesión de los usuarios?
   - **RESPUESTA:** el cliente

Un cliente utiliza Amazon S3 para almacenar datos. El cliente configuró una nube virtual privada con Amazon VPC y está ejecutando un servidor web en una instancia EC2 en la VPC. El cliente configuró una gateway de Internet como parte de la VPC para que se pueda obtener acceso al servidor web mediante la consola de administración de AWS o la interfaz de línea de comandos de AWS (CLI de AWS). Cuando el cliente utiliza la CLI de AWS, la conexión requiere el uso de claves de Secure Shell (SSH).



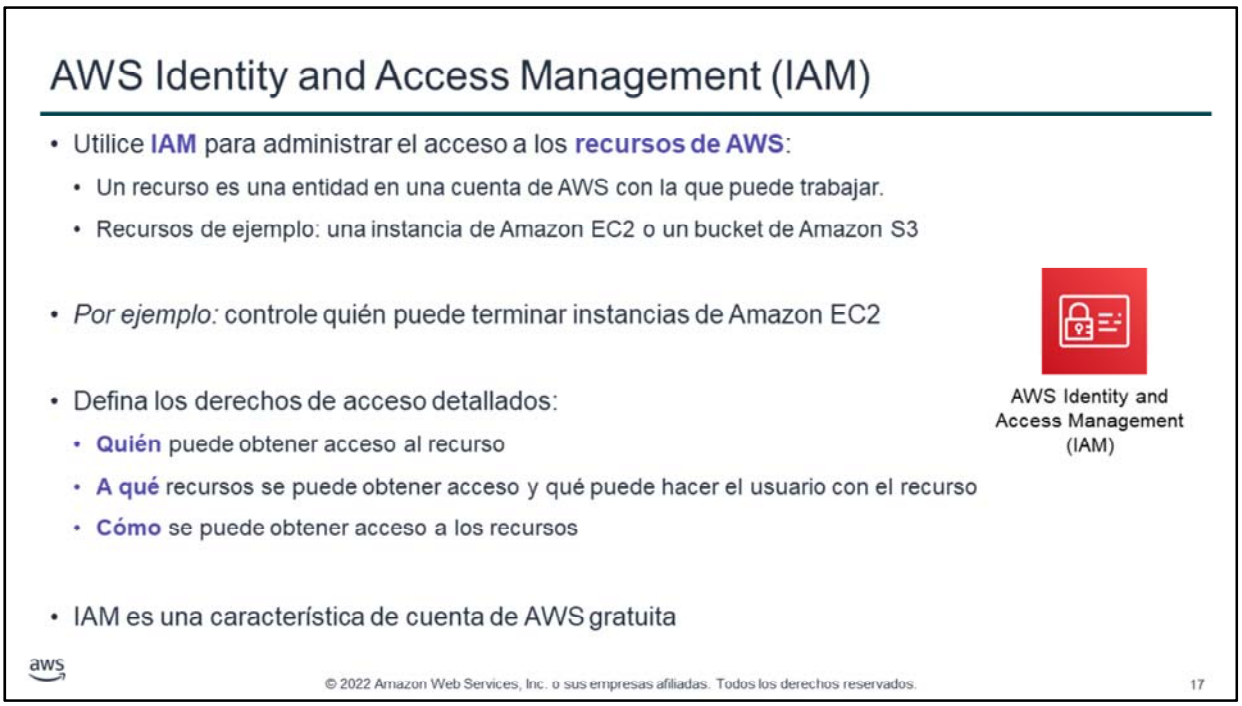
Estos son algunos de los aprendizajes clave de esta sección del módulo:

- AWS y el cliente comparten responsabilidades en materia de seguridad:
    - AWS es responsable de la seguridad **de** la nube.
    - El cliente es responsable de la seguridad **en** la nube.

- **AWS es responsable de proteger la infraestructura** (incluido el hardware, el software, las redes y las instalaciones) que ejecuta los servicios en la nube de AWS.

- En el caso de los servicios clasificados como infraestructura como servicio (IaaS), el **cliente es responsable de realizar las tareas de configuración y administración de seguridad necesarias**.
    - Por ejemplo, actualizaciones del sistema operativo invitado y configuraciones de parches de seguridad, firewall y grupos de seguridad.

Introducción de la sección 2: AWS Identity and Access Management (o IAM)



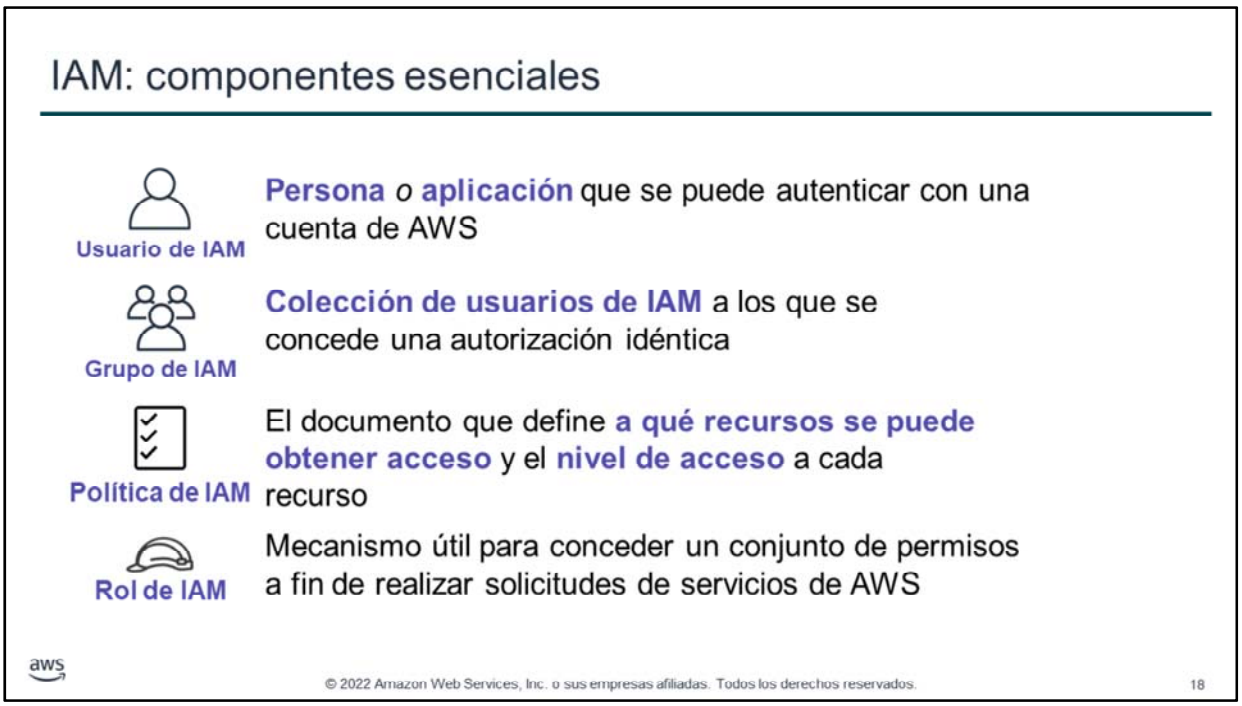
AWS Identity and Access Management (IAM) le permite controlar el acceso a servicios de informática, almacenamiento, base de datos y aplicaciones en la nube de AWS. IAM se puede utilizar para gestionar la autenticación y para especificar y aplicar políticas de autorización a fin de que usted especifique qué usuarios pueden obtener acceso a cada servicio.

IAM es una herramienta que administra el acceso de forma centralizada para lanzar, configurar, administrar y terminar recursos en su cuenta de AWS. Proporciona un control minucioso sobre el acceso a los recursos, incluida la posibilidad de especificar exactamente qué llamadas a la **API** está autorizado a realizar el usuario a cada servicio. Independientemente de si utiliza la consola de administración de AWS, la CLI de AWS o los kits de desarrollo de software (SDK) de AWS, cada llamada a un servicio de AWS es una llamada a la API.

Con IAM, puede administrar *quién* puede obtener acceso a *qué* recursos y *cómo* se puede obtener acceso a ellos. Puede conceder diferentes permisos a diferentes personas para diferentes recursos. Por ejemplo, puede permitir que algunos usuarios tengan acceso completo a Amazon EC2, Amazon S3, Amazon DynamoDB, Amazon Redshift y otros servicios de AWS. Sin embargo, para otros usuarios, puede permitir el acceso de solo lectura a solo unos cuantos buckets de S3. Del mismo modo, puede conceder permiso a otros usuarios para que solamente administren instancias EC2

específicas. También puede permitir que algunos usuarios obtengan acceso únicamente a la información de facturación de la cuenta, pero nada más.

IAM es una característica de su cuenta de AWS y se ofrece sin cargos adicionales.



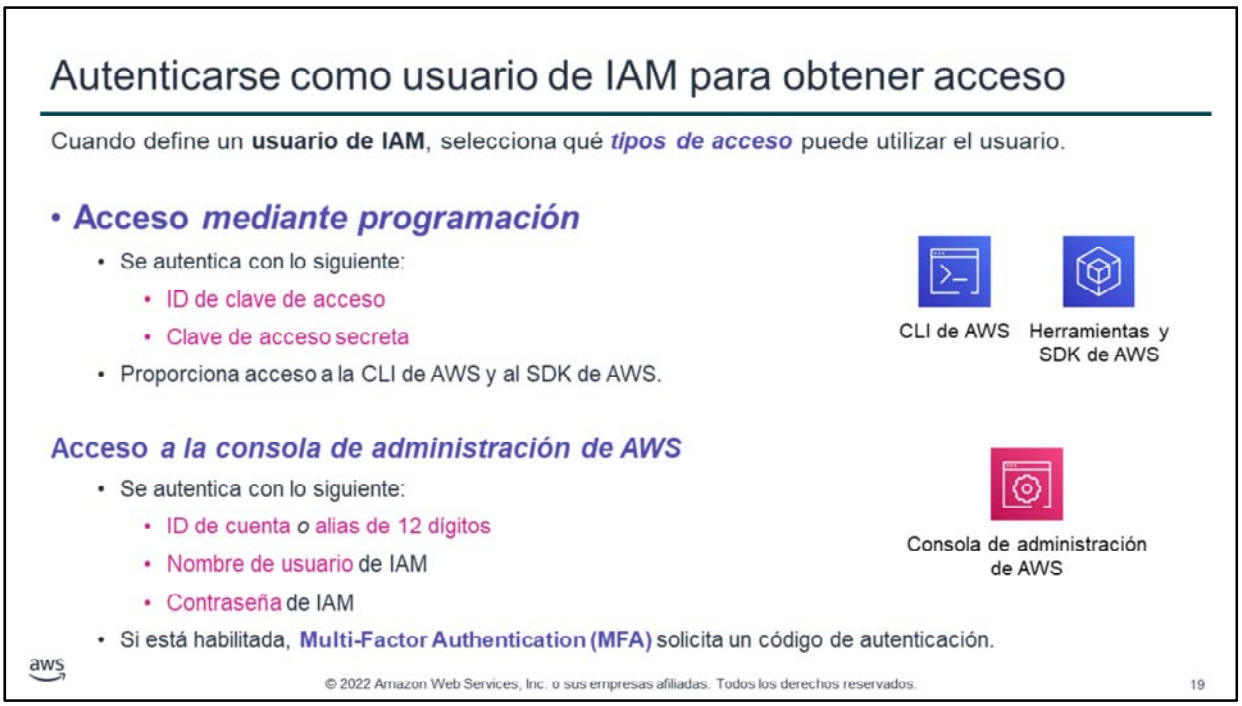
Para comprender cómo utilizar IAM a fin de proteger su cuenta de AWS, es importante comprender el rol y la función de cada uno de los cuatro componentes de IAM.

Un **usuario de IAM** es una persona o aplicación definida en una cuenta de AWS que debe realizar llamadas a la API para los 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 de usuario raíz de la cuenta de AWS. Cada usuario está definido en una única cuenta de AWS.

Un **grupo de IAM** es un conjunto de usuarios de IAM. Puede utilizar grupos de IAM para simplificar la especificación y administración de permisos de varios usuarios.

Una **política de IAM** es un documento que define permisos para determinar lo que los usuarios pueden hacer en la cuenta de AWS. Una política normalmente concede acceso a recursos determinados y especifica lo que el usuario puede hacer con esos recursos. Las políticas también pueden denegar explícitamente el acceso.

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



La **autenticación** es un concepto básico de seguridad informática: un usuario o sistema debe demostrar primero su identidad. Considere cómo se autentica usted mismo cuando va al aeropuerto y desea pasar por el área de seguridad del aeropuerto para poder tomar su vuelo. En esta situación, debe presentar algún tipo de identificación al oficial de seguridad para demostrar quién es usted antes de poder entrar a un área restringida. Un concepto similar se aplica para obtener acceso a los recursos de AWS en la nube.

Cuando define un usuario de IAM, selecciona qué tipo de acceso puede utilizar el usuario para obtener acceso a los recursos de AWS. Hay dos tipos diferentes de acceso que puede asignar a los usuarios: acceso mediante programación y acceso a la consola de administración de AWS. Puede asignar solo acceso mediante programación, solo acceso a la consola o puede asignar ambos tipos de acceso.

Si concede **acceso mediante programación**, el usuario de IAM deberá presentar un **ID de clave de acceso** y una **clave de acceso secreta** cuando realice una llamada a la API de AWS mediante la CLI de AWS, el SDK de AWS o cualquier otra herramienta de desarrollo.

Si concede **acceso a la consola de administración de AWS**, el usuario de IAM deberá completar los campos que aparecen en la ventana de inicio de sesión del navegador. Se le pedirá al usuario que proporcione el ID de cuenta de 12 dígitos o el alias de cuenta

correspondiente. El usuario también debe escribir su nombre de usuario y contraseña de IAM. Si **Multi-Factor Authentication (MFA)** está habilitada para el usuario, también se le solicitará un código de autenticación.

MFA de IAM

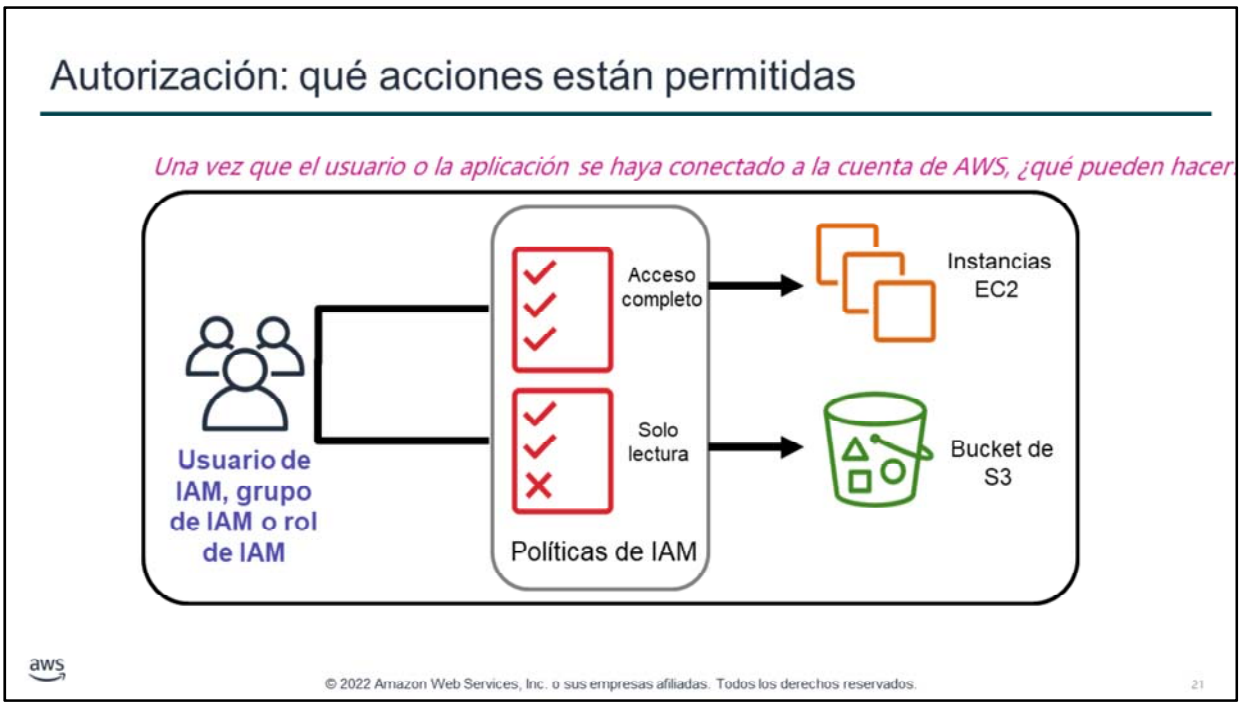
- MFA proporciona más seguridad.
- Además del **nombre de usuario** y la **contraseña**, MFA requiere un código de autenticación único para acceder a los servicios de AWS.

Consola de administración de AWS

Se puede obtener acceso a los servicios y recursos de AWS mediante la consola de administración de AWS, la CLI de AWS o los SDK y las API. Para mayor seguridad, recomendamos habilitar MFA.

Con MFA, los usuarios y los sistemas deben proporcionar un **token de MFA** (además de las credenciales de inicio de sesión habituales) para poder obtener acceso a los servicios y recursos de AWS.

Las opciones para generar el token de autenticación de MFA incluyen **aplicaciones virtuales conformes con MFA** (como Google Authenticator o Authy 2-Factor Authentication), **dispositivos de clave de seguridad U2F** y **dispositivos MFA físicos**.



La **autorización** es el proceso de determinar qué permisos debe concederse a un usuario, servicio o aplicación. Una vez que un usuario ha sido autenticado, debe estar autorizado para acceder a un servicio de AWS.
De forma predeterminada, los usuarios de IAM no tienen permiso para obtener acceso a los recursos o los datos en una cuenta de AWS. En su lugar, debe conceder permisos de forma explícita a un usuario, grupo o rol mediante la creación de una *política*, la cual es un documento en formato JavaScript Object Notation (JSON). Una política enumera los permisos que habilitan o deniegan el acceso a los recursos de la cuenta de AWS.

IAM: autorización

- Asigna permisos mediante la creación de una política de IAM.
- Los permisos determinan **qué recursos y operaciones** están permitidas:
  - De forma predeterminada, todos los permisos están denegados implícitamente.
  - Si algo está denegado explícitamente, nunca se permite.

Práctica recomendada: seguir el **principio de mínimo privilegio**.

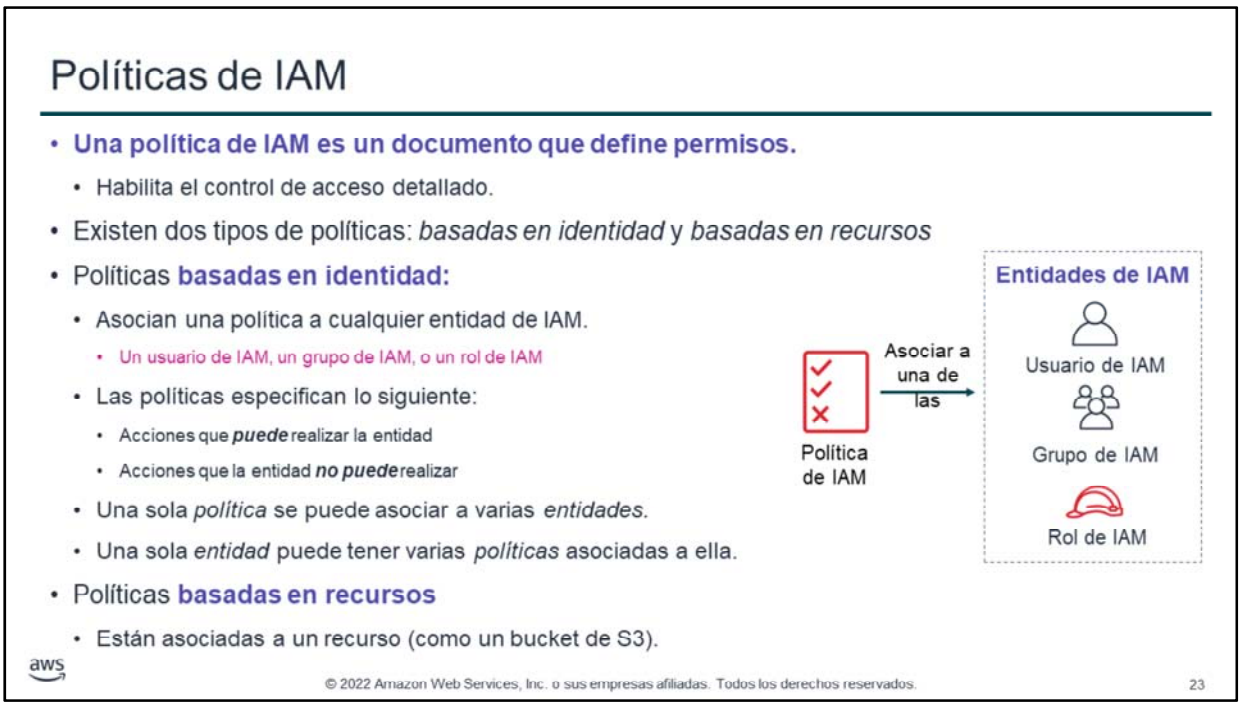
Permisos de IAM

Nota: El alcance de las configuraciones de servicios de IAM es **global**. Las configuraciones se aplican a todas las regiones de AWS.

Para asignar un permiso a un usuario, grupo o rol, debe crear una **política de IAM** (o encontrar una política existente en la cuenta). No existen permisos predeterminados. Todas las acciones de la cuenta se deniegan al usuario de forma predeterminada (*denegación implícita*) a menos que dichas acciones se permitan explícitamente. Se denegarán todas las acciones que no se hayan permitido explícitamente. Cualquier acción que deniegue explícitamente, se denegará en todo momento.

El **principio de mínimo privilegio** es un concepto importante en la seguridad informática. Fomenta que conceda únicamente los privilegios de usuario mínimos que necesita el usuario, en función de las necesidades de los usuarios. Cuando crea políticas de IAM, se recomienda seguir este consejo de seguridad para conceder *privilegios mínimos*. Determine las tareas que tienen que realizar los usuarios y elabore políticas para permitirles realizar *solo* dichas tareas. Comience con un conjunto mínimo de permisos y conceda permisos adicionales según sea necesario. Esto es más seguro que comenzar con permisos que son demasiado amplios y, posteriormente, intentar bloquear los permisos concedidos.

Tenga en cuenta que el alcance de las configuraciones del servicio de IAM es **global**. La configuración no se define en un nivel de región de AWS. Las configuraciones de IAM se aplican en todas las regiones de AWS.



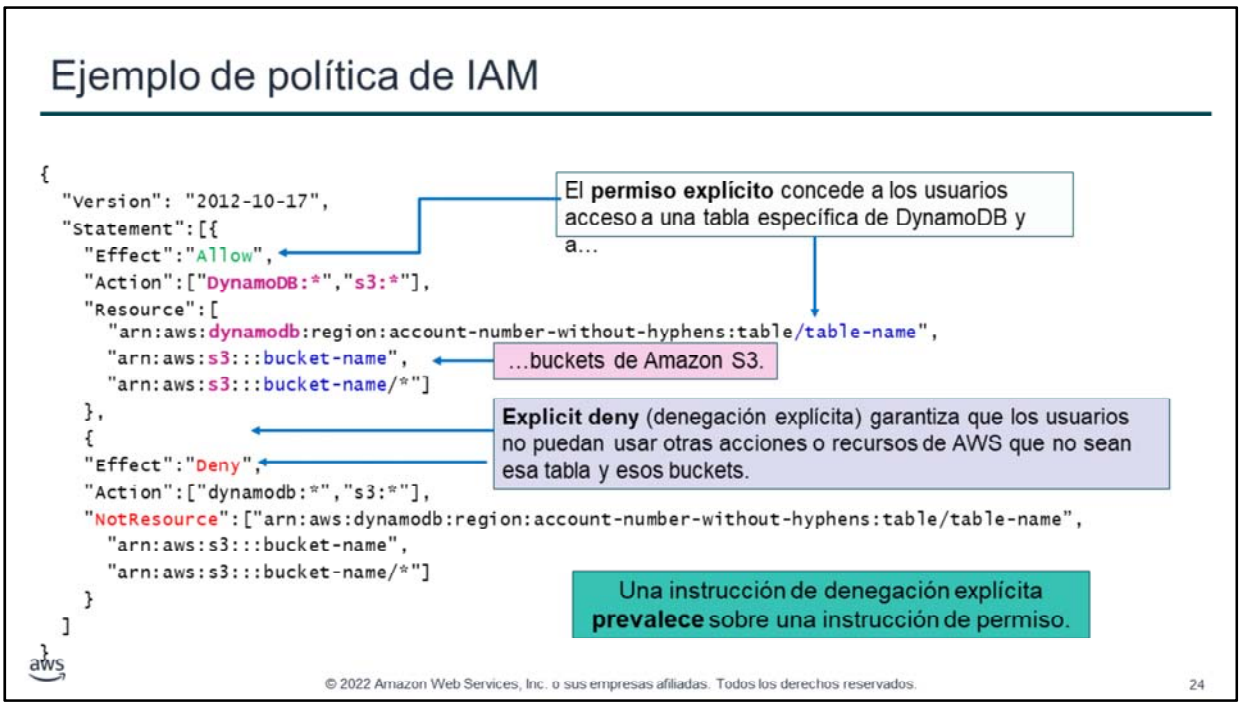
Una política de IAM es una instrucción formal de permisos que se concederá a una entidad. Las políticas se pueden asociar a cualquier entidad de IAM. Las entidades incluyen usuarios, grupos, roles o recursos. Por ejemplo, puede asociar una política a sus recursos de AWS para bloquear todas las solicitudes que no provengan de un rango de direcciones de protocolo de Internet (IP) aprobado. Las políticas especifican cuáles son las acciones permitidas, cuáles son los recursos a los que estas tienen permiso y cuál será el efecto cuando el usuario solicite acceso a los recursos.

El orden en que se evalúan las políticas no modifica el resultado de la evaluación. Se evalúan todas las políticas y el resultado es siempre el permiso o la denegación de la solicitud. Cuando hay un conflicto, se aplica la política más restrictiva.

Hay dos tipos de políticas de IAM. Las **políticas basadas en identidad** son políticas de permisos que puede asociar a una entidad principal (o identidad), como por ejemplo un usuario, rol o grupo de IAM. Estas políticas controlan qué acciones puede realizar dicha identidad, en qué recursos y en qué condiciones. Las políticas basadas en identidad se pueden clasificar del siguiente modo:

- **Políticas administradas:** políticas independientes basadas en identidad que puede asociar a varios usuarios, grupos y roles en su cuenta de AWS.
- **Políticas insertadas:** políticas que crea y administra y que están insertadas directamente en un único usuario, grupo o rol.

Las **políticas basadas en recursos** son documentos de política JSON que puede asociar a un recurso como, por ejemplo, un bucket de S3. Estas políticas controlan qué acciones puede realizar una entidad principal especificada en dicho recurso y en qué condiciones.



Como se ha mencionado anteriormente, los documentos de políticas de IAM se escriben en JSON.

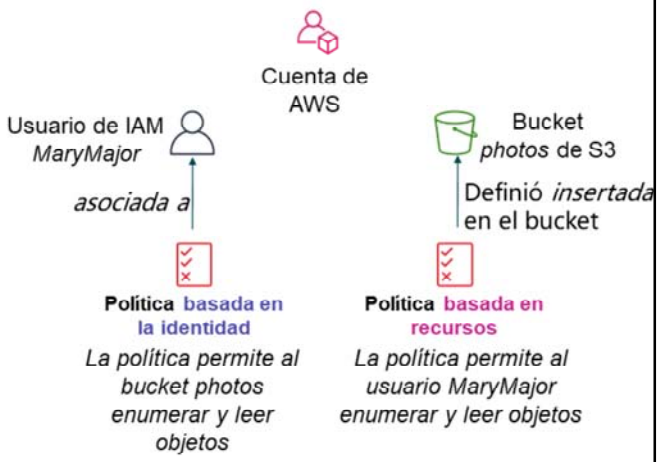
La política de IAM de ejemplo concede a los usuarios acceso únicamente a los siguientes recursos:

- La tabla DynamoDB, cuyo nombre está representado por *table-name*.
- El bucket 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 utilizar ninguna otra acción o recurso de DynamoDB o S3, excepto las acciones y los recursos especificados en la política, aunque se hayan concedido permisos en otra política. Una instrucción de denegación explícita prevalece sobre una instrucción de permiso.

Políticas basadas en recursos

- Las políticas *basadas en identidad* se asocian a un usuario, grupo o rol.
- Las **políticas basadas en recursos** se asocian a un recurso (*no* a un usuario, grupo o rol)
- Características de las políticas basadas en recursos:
  - Especifican quién tiene acceso al recurso y qué acciones se pueden realizar en él.
  - Las políticas son *insertadas* solamente, no se administran.
- Las políticas basadas en recursos solo se admiten en algunos servicios de AWS

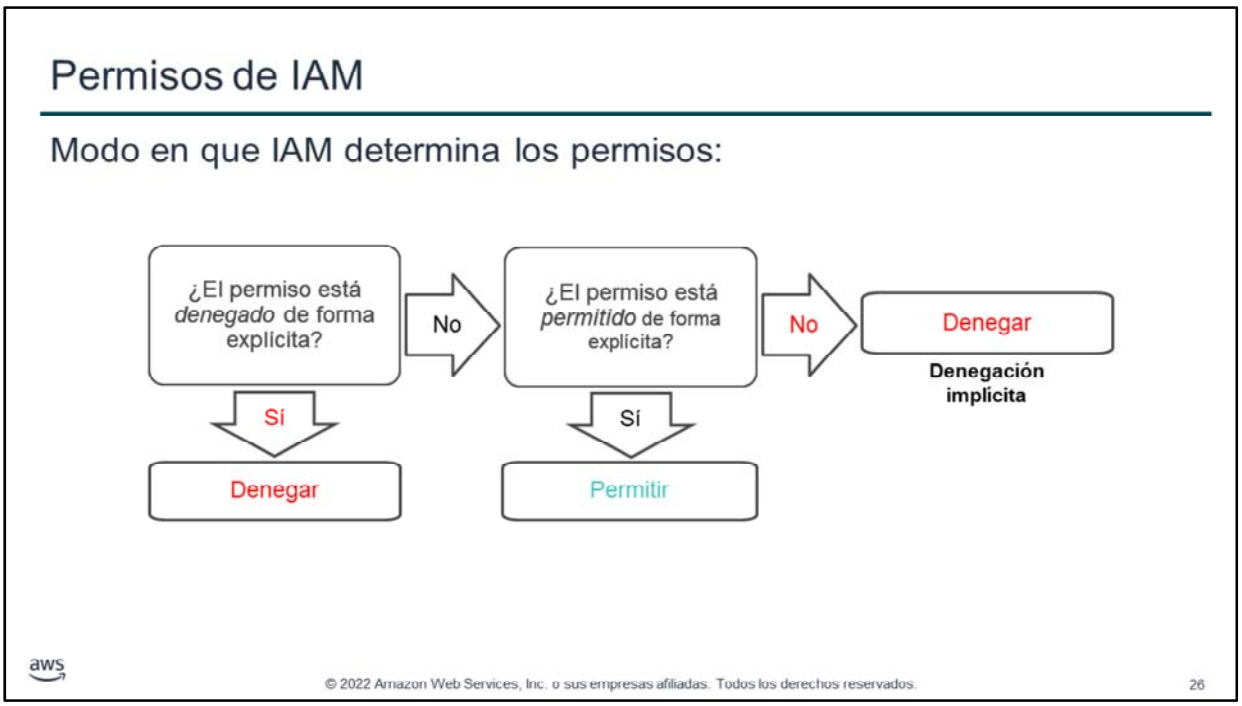


Aunque las políticas basadas en *identidad* están asociadas a un usuario, grupo o rol, las **políticas basadas en recursos** se asocian a un recurso, como un bucket de S3. Estas políticas especifican quién puede obtener acceso al recurso y qué acciones pueden realizar en él.

Las políticas basadas en recursos se definen únicamente de forma **directa,** lo que significa que usted define la política en el propio recurso, en lugar de crear un documento de política de IAM independiente. Por ejemplo, para crear una política de bucket de S3 (un tipo de política basada en recurso) en un bucket de S3, vaya al bucket, haga clic en la pestaña **Permissions** (Permisos), haga clic en el botón **Bucket Policy** (Política de bucket) y defina allí el documento de política con formato JSON. Una lista de control de acceso (ACL) de Amazon S3 es otro ejemplo de una política basada en recursos.

El diagrama muestra dos formas diferentes en las que se podría conceder acceso al usuario *MaryMajor* a objetos en el bucket de S3 denominado *photos.* A la izquierda, puede ver un ejemplo de una política basada en identidad. Una política de IAM que concede acceso al bucket de S3 se asocia al usuario MaryMajor. A la derecha, puede ver un ejemplo de una política basada en recursos. La política de bucket de S3 para el bucket *photos* especifica que el usuario MaryMajor tiene permiso para enumerar y leer los objetos del bucket.

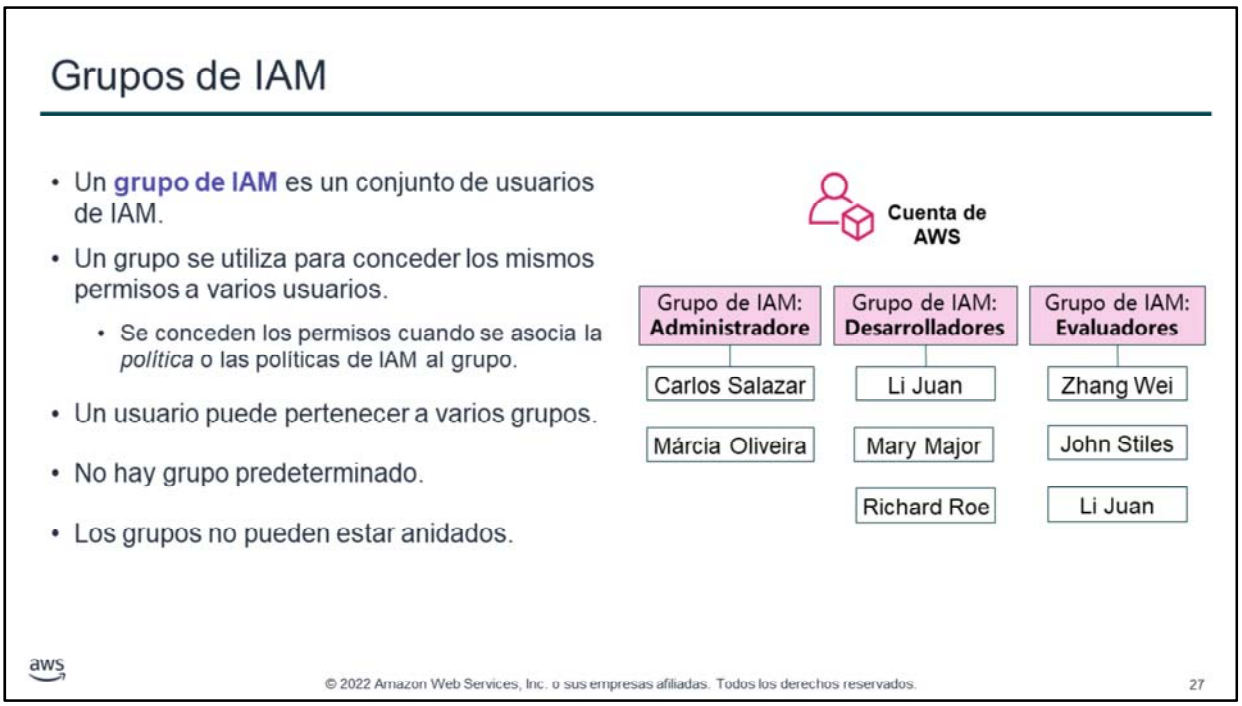
Tenga en cuenta quepodría definir una instrucción de denegación en una política de bucket para restringir el acceso a determinados usuarios de IAM, incluso si se concede acceso a los usuarios en una política basada en identidad independiente. Una instrucción de denegación explícita siempre prevalecerá sobre cualquier instrucción de permiso.



Las políticas de IAM le permiten ajustar los privilegios que se conceden a los usuarios, grupos y funciones de IAM.

Cuando IAM determina si se concede un permiso, IAM comprueba primero la existencia de cualquier **política de denegación explícita** aplicable. Si no existe ninguna denegación explícita, comprueba si existe alguna **política de permisos explícitos** aplicable. Si no existe una política de denegación explícita ni de permiso explícito, IAM vuelve a la forma predeterminada, que consiste en denegar el acceso. Este proceso se denomina **denegación implícita**. El usuario solo podrá realizar la acción si la acción solicitada *no* estádenegada de forma explícita y ***está*** permitida de forma explícita.

Puede ser difícil descubrir si el acceso a un recurso se concederá a una entidad de IAM cuando desarrolle políticas de IAM. El simulador de políticas de IAM es una herramienta útil para probar y solucionar problemas de políticas de IAM.



Un **grupo de IAM** es un conjunto de usuarios de IAM. Los grupos de IAM ofrecen una forma conveniente para especificar permisos a un conjunto de usuarios, lo que puede facilitar la administración de permisos para dichos usuarios.

Por ejemplo, puede crear un grupo de IAM denominado *Desarrolladores* y asociar una política de IAM o varias políticas de IAM al grupo Desarrolladores que concedan los permisos de acceso a los recursos de AWS que normalmente necesitan dichos desarrolladores. Cualquier usuario que agregue al grupo Desarrolladores tendrá automáticamente los permisos asignados al grupo. En tal caso, no es necesario asociar la política de IAM o las políticas de IAM directamente al usuario. Si un nuevo usuario se une a su organización y se le deben conceder privilegios de desarrollador, solo tiene que agregar a ese usuario al grupo Desarrolladores. Del mismo modo, si una persona cambia de trabajo en su organización, en lugar de editar los permisos de ese usuario, simplemente debe eliminar al usuario del grupo.

Las características importantes de los grupos de IAM son las siguientes:
- Un grupo puede contener muchos usuarios y un usuario puede pertenecer a varios grupos.
- Los grupos no pueden estar anidados. Un grupo solo puede contener usuarios y, a su vez, un grupo no puede contener otros grupos.
- No hay ningún grupo predeterminado que incluya automáticamente a todos los

usuarios de la cuenta de AWS. Si desea tener un grupo con todos los usuarios de la cuenta, debe crear el grupo y agregar cada usuario nuevo.

Roles de IAM

- Un **rol de IAM** es una identidad de IAM con permisos específicos.
- Es similar a un usuario de IAM
  - Asocia políticas de permisos a él.
- Es diferente a un usuario de IAM.
  - No está asociado de forma exclusiva a una persona.
  - Está diseñado para *que lo pueda asumir* una **persona**, una **aplicación** o un **servicio**.
- El rol proporciona credenciales de seguridad *temporales*.
- Ejemplos de cómo se utilizan los roles de IAM para **delegar** el acceso:
  - Utilizado por un usuario de IAM en la misma cuenta de AWS que utiliza el rol
  - Utilizado por un servicio de AWS, como Amazon EC2, en la misma cuenta que utiliza el rol
  - Utilizado por un usuario de IAM en una cuenta de AWS diferente a la que utiliza el rol

Rol de IAM

Un **rol de IAM** es una identidad de IAM que se puede crear en su cuenta y que tiene permisos específicos. Un rol de IAM es **similar a un usuario de IAM** porque también es una identidad de AWS a la que puede asociar políticas de permisos y esos permisos determinan lo que la identidad puede hacer y lo que no en AWS. Sin embargo, en lugar de estar asociada únicamente a una persona, el objetivo es que pueda asignarse un rol a cualquier persona que lo necesite. Además, un rol no tiene asociadas credenciales estándar a largo plazo, como una contraseña o claves de acceso. En su lugar, cuando se asume un rol, este le proporciona credenciales de seguridad temporales para la sesión de rol.

Puede **utilizar roles para delegar el acceso a usuarios, aplicaciones o servicios** que normalmente no tendrían acceso a los recursos de AWS. Por ejemplo, es posible que desee conceder acceso a los usuarios de su cuenta de AWS a los recursos que no suelen tener o conceder acceder a los usuarios de una cuenta de AWS a los recursos de otra cuenta. También puede que quiera permitir que una aplicación móvil utilice los recursos de AWS, pero no desea integrar las claves de AWS en la aplicación (donde serían difíciles de rotar y donde los usuarios pueden potencialmente extraerlas y usarlas de forma incorrecta). Además, a veces es posible que desee conceder acceso a AWS a los usuarios que ya tienen identidades definidas fuera de AWS, como en su directorio corporativo. O bien, es posible que quiera conceder acceso a su cuenta a terceros para que puedan realizar una auditoría en los recursos.

En todos estos casos de uso de ejemplo, los roles de IAM son un componente esencial en la implementación en la nube.

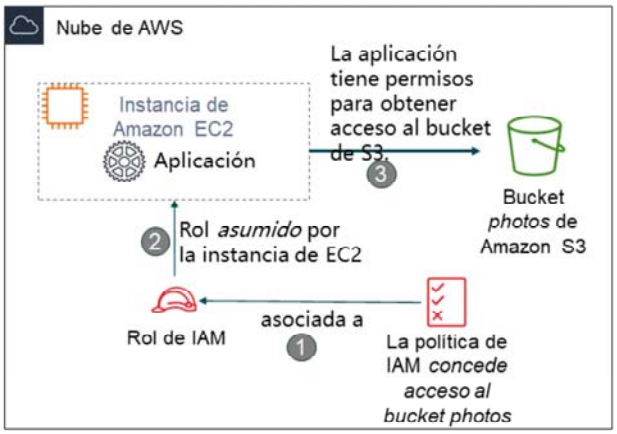
Ejemplo de uso de un rol de IAM

Situación:

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

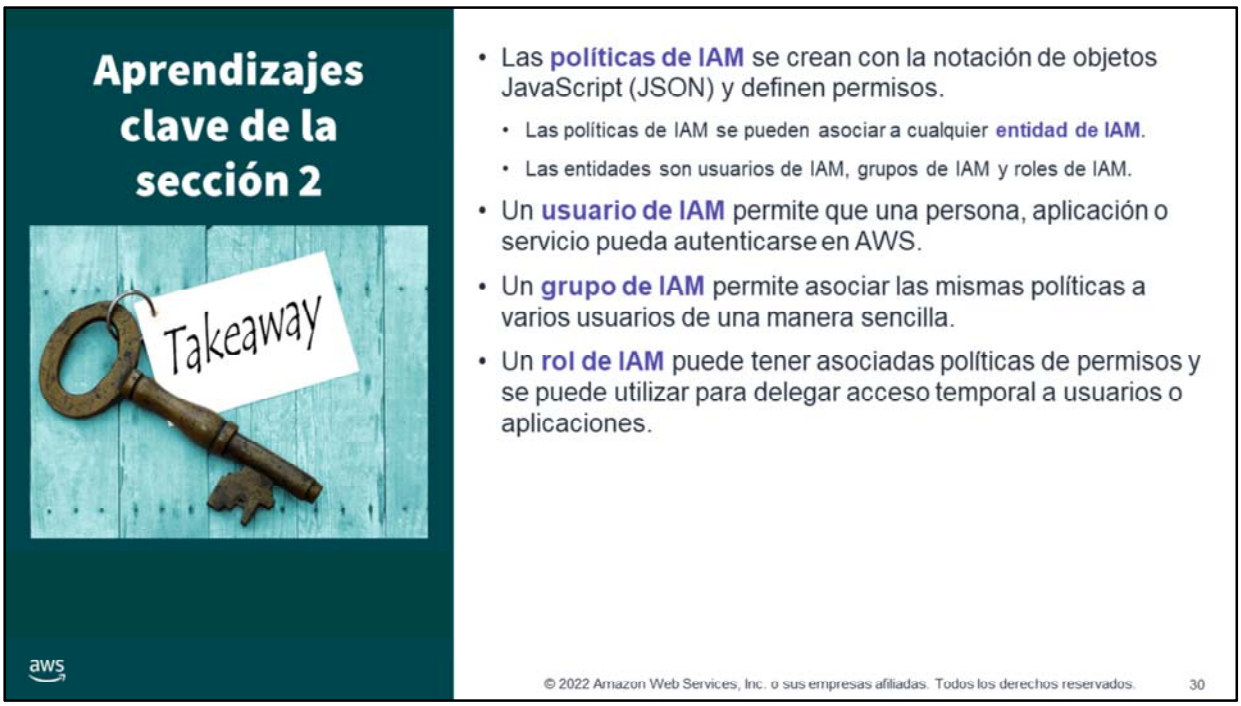
Solución:

- Definir una política de IAM que conceda acceso al bucket de S3
- Asociar la política a un rol
- Permitir que la instancia EC2 asuma el rol



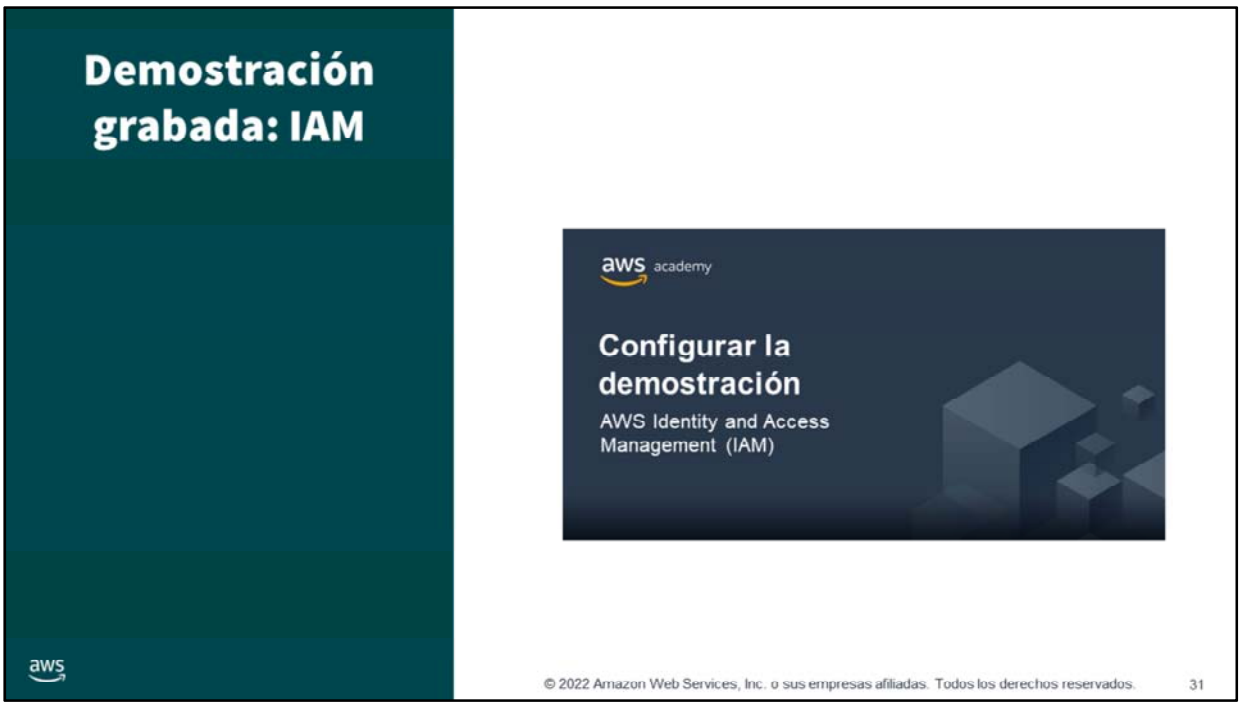
En el diagrama, un desarrollador ejecuta una aplicación en una instancia EC2 que requiere acceso al bucket de S3 denominado *photos*. Un administrador crea el rol de IAM y lo asocia a la instancia EC2. El rol incluye una política de permisos que otorga acceso de solo lectura al bucket de S3 especificado. También incluye una política de confianza que permite a la instancia EC2 asumir el rol y obtener las credenciales temporales. Cuando la aplicación se ejecuta en la instancia, puede utilizar las credenciales temporales del rol para obtener acceso al bucket **photos**. El administrador no necesita conceder permiso al desarrollador de la aplicación para obtener acceso al bucket photos y el desarrollador nunca necesita compartir ni administrar las credenciales.

Para obtener más información acerca de este ejemplo, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias de Amazon EC2.



Estos son algunos de los aprendizajes clave de esta sección del módulo:

Las **políticas de IAM** se crean con la notación de objetos JavaScript (JSON) y definen permisos.
    - Las políticas de IAM se pueden asociar a cualquier **entidad de IAM**.
    - Las entidades son usuarios de IAM, grupos de IAM y roles de IAM.
- Un **usuario de IAM** permite que una persona, aplicación o servicio pueda autenticarse en AWS.
- Un **grupo de IAM** permite asociar las mismas políticas a varios usuarios de una manera sencilla.
- Un **rol de IAM** puede tener asociadas políticas de permisos y se puede utilizar para delegar acceso temporal a usuarios o aplicaciones.

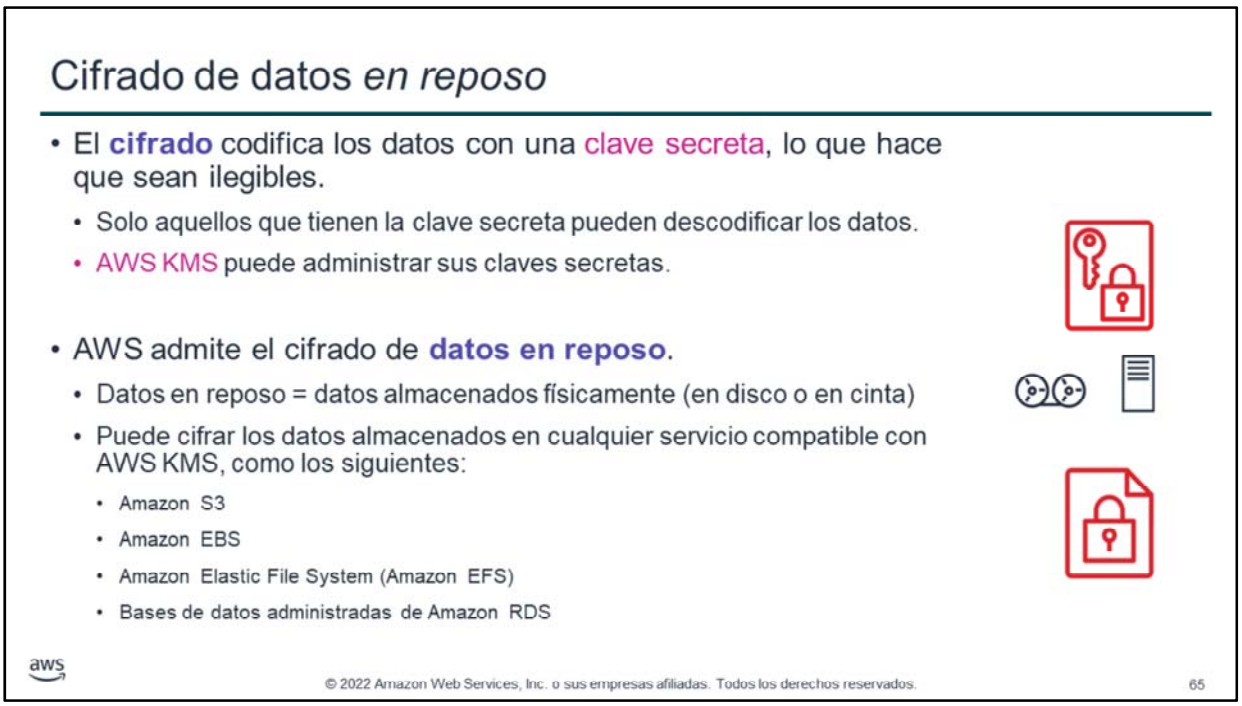


Ahora, dedique un momento a ver la demostración de IAM. La grabación dura poco más de 4 minutos y refuerza muchos de los conceptos que se han tratado en esta sección del módulo.

En la demostración, se muestra cómo configurar los siguientes recursos mediante la consola de administración de AWS:

- Un rol de IAM que utilizará una instancia EC2
- Un grupo de IAM
- Un usuario de IAM

Presentación de la sección 5: Protección de datos en AWS



El **cifrado de datos** es una herramienta esencial cuando desea proteger los datos digitales. El cifrado de datos toma datos legibles y los codifica de forma que sean ilegibles para cualquiera que no tenga acceso a la clave secreta y quiera utilizarla para decodificarlos. Por lo tanto, aunque un atacante obtenga acceso a sus datos, no podrá entenderlos.

Los datos en reposo se refieren a datos que se almacenan físicamente en disco o en cinta.

Puede crear sistemas de archivos cifrados en AWS para que todos sus datos y metadatos se cifren en reposo mediante el algoritmo de cifrado estándar y abierto Advanced Encryption Standard (AES) de 256 bits. Cuando utiliza AWS KMS, el cifrado y el descifrado se gestionan de forma automática y transparente, por lo que no es necesario modificar sus aplicaciones. Si su organización está sujeta a políticas corporativas o normativas que requieren el cifrado de datos y metadatos en reposo, AWS recomienda que habilite el cifrado en todos los servicios que almacenan sus datos. Puedecifrar los datos almacenados en cualquier servicio compatible con AWS KMS. Consulte cómo los servicios de AWS utilizan AWS KMS para obtener una lista de los servicios admitidos.

Los **datos en tránsito** se refieren a los datos que se mueven a través de la red. El cifrado de los datos en tránsito se realiza mediante el uso de la seguridad de Transport Layer Security (TLS) 1.2 con un cifrado AES de 256 bits estándar abierto. TLS anteriormente se denominaba capa de conexión segura (SSL).

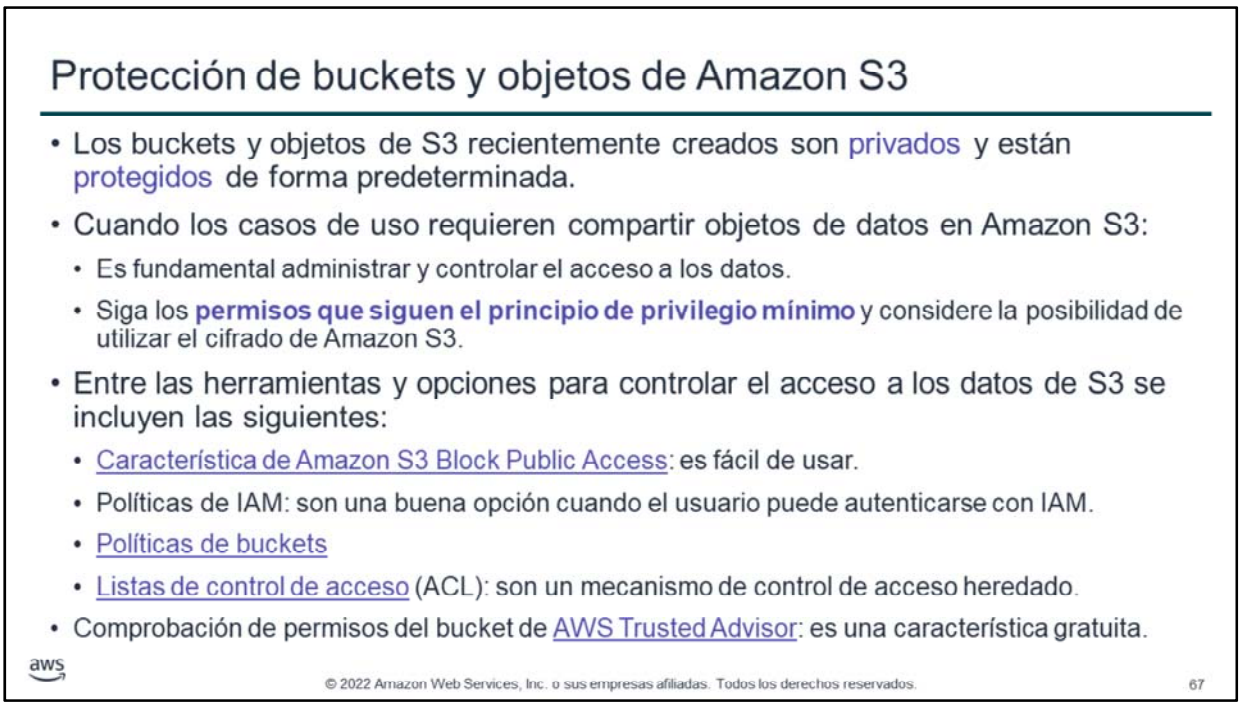
AWS Certificate Manager es un servicio que le permite aprovisionar, administrar e implementar certificados SSL o TLS para su uso con los servicios de AWS y sus recursos internos conectados. Los certificados de SSL o TLS se usan para proteger las comunicaciones por red y para definir la identidad de sitios web mediante Internet y recursos en redes privadas. Con AWS Certificate Manager, puede solicitar un certificado y, luego, implementarlo en recursos de AWS (como balanceadores de carga o distribuciones de CloudFront). AWS Certificate Manager también se encarga de renovar certificados.

El tráfico web que se ejecuta a través de HTTP no es seguro. Sin embargo, el tráfico que se ejecuta a través de **HTTP seguro (HTTPS)** se cifra mediante TLS o SSL. El tráfico HTTPS está protegido contra ataques de acceso no autorizados y ataques de intermediario debido al cifrado bidireccional de la comunicación.

Los servicios de AWS admiten el cifrado de datos en tránsito. Se muestran dos ejemplos de cifrado para datos en tránsito. El primer ejemplo muestra una instancia EC2 que ha

montado un sistema de archivos compartidos de Amazon EFS. Todo el tráfico de datos entre la instancia y Amazon EFS se cifra mediante TLS o SSL. Para obtener más información acerca de esta configuración, consulte Cifrado de datos de EFS en tránsito.

El segundo ejemplo muestra el uso de **AWS Storage Gateway,** un servicio de almacenamiento en la nube híbrida que proporciona acceso en las instalaciones al almacenamiento en la nube de AWS. En este ejemplo, la gateway de almacenamiento está conectada a través de Internet a Amazon S3 y la conexión cifra los datos en tránsito.



De forma predeterminada, todos los buckets de Amazon S3 son privados y *solo* pueden acceder los usuarios a los que se les concede acceso explícitamente. Es fundamental administrar y controlar el acceso a los datos de Amazon S3. AWS proporciona muchas herramientas y opciones para controlar el acceso a sus buckets u objetos de S3, entre los que se incluyen los siguientes:

- El uso de **Amazon S3 Block Public Access.** Esta configuración anula cualquier otra política o permisos de objetos. Habilite **Block Public Access** para todos los buckets que no desee que sean accesibles públicamente. Esta característica proporciona un método sencillo para evitar la exposición no deseada de datos de Amazon S3.

- La escritura de **políticas de IAM** que especifiquen los usuarios o roles que pueden obtener acceso a buckets y objetos específicos. Este método se ha tratado en detalle anteriormente en este módulo.

- La escritura de **políticas de bucket** que definan el acceso a buckets u objetos específicos. Esta opción se suele utilizar cuando el usuario o el sistema no pueden autenticarse mediante IAM. Las políticas de bucket se pueden configurar para conceder acceso entre cuentas de AWS o para conceder acceso público o anónimo a los datos de Amazon S3. Si se utilizan políticas de bucket, deben escribirse detenidamente y probarse en su totalidad. Puede especificar una instrucción de

denegación en una política de bucket para restringir el acceso. El acceso estará restringido incluso si los usuarios tienen permisos concedidos en una política basada en identidad asociada a los usuarios.

- Configuración de **listas de control de acceso (ACL)** en sus buckets y objetos. Las ACL se utilizan con menos frecuencia (las ACL preceden a la IAM). Si utiliza ACL, no establezca un acceso demasiado abierto o permisivo.

- **AWS Trusted Advisor** proporciona una característica de comprobación de permisos de buckets, que es una herramienta útil para descubrir si alguno de los buckets de su cuenta tiene permisos que conceden acceso global.

AWS Identity and Access Management (IAM)

¿QUÉ ES?

Es un sistema de seguridad para identificarse.
El principal rol es saber qué personas y máquinas son las correctas a las que acceden a esos recursos.
Es un servicio gratuito.

EJEMPLO DE UTILIDAD:

En una empresa con diferentes usuarios y actividades, permite crear usuarios y contraseñas sin que se compartan los recursos entre dichos usuarios.

¿QUÉ ES?

Es un servidor de administración de cuentas que permite administrar de forma centralizada varias cuentas de AWS.
Organiza las cuentas, políticas de control de servicios, unifica la facturación y las unidades organizativas.

EJEMPLO DE UTILIDAD:

Un equipo de seguridad tiene control sobre toda la empresa, mientras que los equipos de desarrollo tienen unos límites definidos por el equipo de seguridad.

AWS Key Management Service (AWS KMS)

¿QUÉ ES?

Es un servicio que permite crear, administrar y controlar llaves de cifrado para proteger datos en la nube.
Protege datos en la nube.

EJEMPLO DE UTILIDAD:

Para cifrar datos almacenados en diferentes servicios.
Se puede controlar quién puede usar las llaves aplicando permisos de auditoria.

Amazon Cognito

¿QUÉ ES?

Gestiona identidades y control de acceso para aplicaciones web y móviles, permitiendo autenticar usuarios, federar identidades externas y proporcionar credenciales seguras para acceder a recursos AWS.

Integra autenticación mediante proveedores externos (Google, FB o Apple).

Emite tokens y credenciales temporales.

Aplica políticas de seguridad como MCA.

EJEMPLO DE UTILIDAD:

Tienda online donde haya un login por usuario y contraseña.

¿QUÉ ES?

Es un servicio de protección contra ataques DDOS.
Su función es detectar ataques que intenten saturar la infraestructura de AWS

EJEMPLO DE UTILIDAD:

Proteger una tienda online para que no te “tumben” la página web.
Protege plataformas con muchos usuarios.

¿QUÉ ES?

Es un servicio que registra y evalúa continuamente los recursos de AWS para ayudar a las organizaciones a evaluar sus configuraciones de forma automática .

Registra configuraciones, evalúa el cumplimiento, detecta cambios y verifica relaciones.

EJEMPLO DE UTILIDAD:

En la gobernanza en la nube, para corregir errores.
Una empresa que utilice datos sensibles de clientes, este servicio revisa en tiempo real si todos cifrados o no.
Si no están cifrados o hay algún error lo cifra automáticamente.

¿QUÉ ES?

Tiene dos secciones: informes para el cumplimiento de la seguridad y normativa
Acuerdos para revisar, aceptar acuerdos entre organizaciones, el tratamiento de datos y ayuda a garantizar que se cumplen los requisitos legales.

EJEMPLO DE UTILIDAD:

Empresa que necesite cumplir una auditoria con unos requisitos legales muy fijados. Por ej ISO 9001.

¿QUÉ ES?

Permite crear y administrar catálogos de servicios aprobados, bases de datos o aplicaciones.

EJEMPLO DE UTILIDAD:

Restaurante con un menú fijo, harías el catálogo del menú.

¿QUÉ ES?

Analiza tus buckets de S3 para clasificar y proteger datos. Detecta información sensible como DNI´s, las identifica si son de riesgo y los clasifica por niveles de sensibilidad y envía alertas al encontrar algún problema.

EJEMPLO DE UTILIDAD:

Empresa con archivos CSV con los datos de los clientes sensibles.

¿QUÉ ES?

Analiza los recursos para encontrar vulnerabilidades, configuraciones inseguras y problemas de cumplimiento. Escanea instancias EC2 y contenedores, evalúa las configuraciones que pueden ser explotadas y prioriza riesgos según la prioridad de ser aprovechados por atacantes.

EJEMPLO DE UTILIDAD:

Un PC con un SW anticuado, al que le falten parches de seguridad.

¿QUÉ ES?

Detecta amenazas, monitoriza tu entorno AWS, identifica actividad maliciosa, supervisa los registros de eventos. Da alertas detalladas y los clasifica por severidad ante amenazas.

EJEMPLO DE UTILIDAD:

Detecta comunicaciones entre un servidor externo y tu servicio y las soluciona.