Módulo 1: Introducción a AWS

Introducción

1.1 Introducción al módulo (resumen vídeo).

El curso ofrece información básica sobre AWS, sus beneficios y cómo puede mejorar la gestión empresarial. AWS proporciona servicios que abarcan desde computación, almacenamiento y seguridad, hasta blockchain, inteligencia artificial y machine learning. También permite alquilar herramientas especializadas, como sistemas de administración y satélites.

Se explica el modelo cliente-servidor con una metáfora de una cafetería:

- El cliente realiza una solicitud.
- El servidor (simbolizado por una camarera) procesa y responde.
- En AWS, este servidor sería una instancia EC2.

Pagar solo por lo que se usa es uno de los principios clave de AWS, comparado con pagar empleados solo por las horas trabajadas. AWS permite escalar o reducir capacidad según las necesidades sin pagos anticipados ni limitaciones físicas, optimizando recursos y costos.

Finalmente, el curso busca preparar a los participantes para obtener la certificación Cloud Practitioner.

1.2 ¿Qué es un modelo cliente servidor?

El modelo cliente-servidor es un diseño básico en computación donde dos entidades interactúan:

- **Cliente**: Puede ser un navegador web, una aplicación móvil o de escritorio que permite a los usuarios realizar solicitudes.
- **Servidor**: Es un sistema que procesa las solicitudes del cliente y proporciona respuestas. Un ejemplo es **Amazon EC2**, un servidor virtual en AWS.

Ejemplo:

1. El cliente solicita algo, como un artículo de noticias, un resultado deportivo o un vídeo.
2. El servidor recibe esta solicitud, la evalúa y devuelve la información al cliente.

Este modelo es la base de gran parte de la tecnología actual, desde aplicaciones web hasta servicios en la nube.

Computación en la nube

1.1 ¿Qué es la informática en la nube? (resumen vídeo).

La computación en la nube se define como la **entrega bajo demanda de recursos de TI a través de Internet** con un modelo de pago por uso. AWS ofrece flexibilidad al permitir a las empresas acceder rápidamente a recursos como servidores virtuales o almacenamiento masivo, sin necesidad de solicitarlos ni administrarlos previamente. Cuando ya no se necesitan, se pueden liberar inmediatamente, dejando de generar costos.

Conceptos clave:

1. **Entrega bajo demanda**: Los recursos están disponibles cuando se necesitan y se liberan cuando ya no son requeridos, eliminando la rigidez de los centros de datos tradicionales.
2. **Trabajos pesados no diferenciados de las TI**: AWS se encarga de tareas comunes y repetitivas, como administrar bases de datos, para que las empresas puedan centrarse en lo que las hace únicas, como los datos o las estructuras específicas.
3. **Modelo de pago por uso**: Similar a ajustar el personal de una tienda a las horas de mayor actividad, AWS permite pagar solo por los recursos utilizados en el momento necesario, evitando costos innecesarios en periodos de inactividad, como fines de semana.

AWS ayuda a las empresas a reducir complejidades y enfocarse en sus fortalezas competitivas, eliminando la necesidad de administrar infraestructuras propias.

1.2 Modelos de despliegue de computación en la nube.

Al elegir una estrategia en la nube, las empresas deben considerar factores como los requisitos de las aplicaciones, herramientas de administración y necesidades de infraestructura existente. Existen tres modelos principales de despliegue:

1.Despliegue basado en la nube:

El despliegue basado en la nube implica ejecutar, migrar o diseñar aplicaciones directamente en la nube. Las aplicaciones pueden:

- Ser migradas desde infraestructuras existentes.
- Diseñarse desde cero para aprovechar la nube.

Las empresas pueden optar por usar:

- Infraestructura de bajo nivel: Requiere administración directa por parte del personal de TI.
- Servicios de nivel superior: Simplifican la gestión, arquitectura y escalado.

Un ejemplo sería crear una aplicación compuesta por servidores virtuales, bases de datos y redes, totalmente gestionada en la nube.

2.Despliegue en las instalaciones:

El despliegue en las instalaciones, también conocido como nube privada, utiliza herramientas de virtualización y administración de recursos para gestionar aplicaciones y aumentar la eficiencia. Este modelo opera desde el centro de datos de la empresa, similar a la infraestructura heredada, pero incorpora tecnologías modernas para optimizar el uso de recursos.

Ejemplo: Aplicaciones ejecutadas en tecnología localizada dentro del centro de datos propio de la empresa.

3.Despliegue híbrido:

El despliegue híbrido combina recursos basados en la nube con infraestructura local, permitiendo integrar aplicaciones heredadas con servicios modernos. Es ideal para situaciones donde:

- Las aplicaciones heredadas funcionan mejor en las instalaciones.
- Las regulaciones requieren mantener ciertos datos localmente.

Ejemplo: Una empresa mantiene aplicaciones heredadas en sus instalaciones mientras utiliza servicios en la nube para procesar y analizar datos automáticamente.

1.3 Beneficios de la computación en la nube.

Pasar de gasto inicial a gasto variable:

En lugar de hacer grandes inversiones en infraestructura, la computación en la nube permite pagar solo por los recursos utilizados, lo que reduce los gastos iniciales.

Deja de gastar dinero en mantener centros de datos:

La nube permite centrarse en aplicaciones y clientes, reduciendo los costos y esfuerzos asociados con la gestión de servidores y centros de dato

Deja de hacer suposiciones sobre la capacidad:

Con la nube, no es necesario predecir la capacidad antes de desplegar una aplicación. Se puede escalar recursos según la demanda y pagar solo por lo usado.

Beneficios de grandes economías de escala:

Los proveedores de nube, como AWS, aprovechan economías de escala, lo que resulta en precios más bajos gracias a la agregación del uso de muchos clientes.

Aumento de la velocidad y agilidad:

La nube facilita el desarrollo y despliegue rápido de aplicaciones, lo que permite experimentar e innovar sin esperar semanas para obtener recursos.

Alcance global en minutos:

La infraestructura global de AWS permite desplegar aplicaciones rápidamente para clientes de todo el mundo, con baja latencia, lo que mejora la experiencia del usuario.

Módulo 2: Computación en la nube

Introducción

1.1 Introducción a Amazon EC2 (resumen vídeo).

Amazon Elastic Compute Cloud (EC2) es un servicio de computación flexible, rentable y rápido que permite a las empresas alquilar servidores virtuales para sus aplicaciones. A diferencia de gestionar servidores físicos, que requiere tiempo, dinero y recursos para su compra, instalación y mantenimiento, EC2 ofrece la capacidad de lanzar instancias virtuales en pocos minutos, pagar solo por lo que se usa y evitar los costos de servidores no utilizados.

EC2 se ejecuta en servidores físicos administrados por AWS con tecnología de virtualización, lo que permite el uso compartido de recursos entre varias instancias virtuales (tenencia múltiple), asegurando la protección y el aislamiento de cada instancia. Además, ofrece control sobre el sistema operativo (Windows o Linux), el software que se ejecuta, el tamaño de la instancia (escalado vertical) y la configuración de la red.

Este servicio facilita la innovación y agilidad empresarial, ya que permite configurar, escalar y gestionar instancias de manera sencilla y económica, sin necesidad de manejar infraestructura física.

1.2 Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud (Amazon EC2) ofrece capacidad de computación segura y escalable en la nube mediante instancias virtuales. A diferencia de los recursos tradicionales, con EC2 no es necesario gastar dinero por adelantado en hardware ni esperar su entrega. Puedes aprovisionar y lanzar una instancia de EC2 en minutos, usarla solo cuando la necesites y pagar solo por el tiempo que esté en ejecución. Esto permite ahorrar costes, ya que solo se paga por la capacidad de servidor utilizada, sin necesidad de mantener recursos innecesarios.

1.3 Como funciona Amazon EC2

1.Iniciar:
Para iniciar una instancia en Amazon EC2, primero debes seleccionar una plantilla que contenga configuraciones básicas, como el sistema operativo y las aplicaciones necesarias. Luego, eliges el tipo de instancia, que define las especificaciones de hardware de la máquina virtual. Además, es importante configurar la seguridad para controlar el tráfico de red que puede acceder a la instancia, lo cual se abordará en más detalle más adelante en el curso.

2.Conexión:
A continuación, te conectas a la instancia. Puedes conectarte a la instancia de varias formas. Tus programas y aplicaciones tienen distintos métodos para conectarse directamente a la instancia e intercambiar datos. Los usuarios también pueden conectarse a la instancia iniciando sesión y accediendo al escritorio del equipo.

3.Uso:
Una vez que te hayas conectado a la instancia, puedes empezar a usarla. Puedes ejecutar comandos para instalar software, añadir almacenamiento, copiar y organizar archivos y mucho más.

Tipos de instancias de Amazon EC2

1.1 Más información sobre Amazon EC2 (resumen vídeo)

En este vídeo, se explica que Amazon EC2 ofrece diferentes tipos de instancias, cada una diseñada para tareas específicas. Al igual que en una cafetería, donde se necesitan empleados con habilidades diversas (cajeros, camareros, expertos en café), AWS proporciona varias familias de instancias optimizadas para distintos propósitos.

1.2 Tipos de instancias de Amazon EC2

Instancias de propósito general: Proporcionan un equilibrio entre recursos de computación, memoria y redes, ideales para aplicaciones como servidores de aplicaciones, servidores de juegos y bases de datos pequeñas y medianas.

Instancias de computación optimizada: Son adecuadas para aplicaciones que requieren alto rendimiento de computación, como servidores web de alto rendimiento, servidores de aplicaciones de uso intensivo y servidores de juegos. También son útiles para cargas de trabajo de procesamiento por lotes.

Instancias con optimización de memoria: Están diseñadas para manejar cargas de trabajo que procesan grandes conjuntos de datos en memoria, como bases de datos de alto rendimiento o aplicaciones que requieren procesamiento en tiempo real de grandes cantidades de datos no estructurados.

Instancias de computación acelerada: Usan aceleradores de hardware (coprocesadores) para realizar funciones de forma más eficiente que el software. Son ideales para cargas de trabajo gráficas, procesamiento de gráficos y streaming de aplicaciones.

Instancias optimizadas para el almacenamiento: Están diseñadas para cargas de trabajo que requieren acceso rápido y secuencial a grandes conjuntos de datos almacenados localmente, como sistemas de archivos distribuidos, aplicaciones de almacenamiento de datos y procesamiento de transacciones en línea de alta frecuencia. Están optimizadas para alto rendimiento de IOPS (operaciones de entrada/salida por segundo).

Precios de Amazon EC2

1.1Amazon EC2 ofrece varias opciones de facturación:

Bajo demanda: Ideal para cargas de trabajo irregulares o a corto plazo. Se paga solo por el tiempo de uso de las instancias, sin costos iniciales ni compromisos. No es recomendado para cargas de trabajo de larga duración debido a que no ofrece los mayores ahorros.

Instancias reservadas: Ofrecen un descuento por comprometerte a un uso durante uno o tres años. Existen dos tipos:

- **Standard**: Se elige tipo, tamaño de instancia y región, garantizando capacidad.
- **Convertibles**: Permiten flexibilidad en el tipo y zona de la instancia, con un descuento mayor.

Savings Plans: Un modelo flexible donde te comprometes a un gasto por hora en una familia de instancias y región. Ahorros de hasta un 72% en comparación con tarifas bajo demanda, sin necesidad de especificar el tipo de instancia o sistema operativo.

Instancias de spot: Utilizan capacidad no usada y ofrecen descuentos de hasta un 90%, pero pueden ser interrumpidas en cualquier momento si la demanda de instancias aumenta. Son ideales para cargas de trabajo con flexibilidad en el tiempo de ejecución.

Hosts dedicados: Servidores físicos dedicados para tu uso exclusivo, ideales para cumplir con requisitos de conformidad o licencias de software. Son los más caros de todas las opciones.

Escalado de Amazon EC2

Parte 1 (Resumen vídeo)

En el vídeo se explica cómo AWS resuelve el dilema de los centros de datos físicos, donde las cargas de trabajo varían constantemente. Los centros de datos tradicionales enfrentan el reto de comprar el hardware adecuado para atender tanto la demanda promedio como los picos de carga, lo que a menudo resulta en recursos infrautilizados o insuficientes.

AWS ofrece escalabilidad y elasticidad, lo que permite aprovisionar recursos de computación según la demanda exacta. Si la carga de trabajo aumenta, AWS ajusta automáticamente los recursos, asegurando que los clientes siempre reciban el servicio adecuado sin desperdiciar recursos.

El ejemplo de la preparación de café ilustra cómo, al igual que un sistema de pedidos, se pueden duplicar las instancias para garantizar la disponibilidad continua. Si una instancia falla, otra puede tomar su lugar sin interrupción, creando un sistema redundante y de alta disponibilidad.

Este enfoque de AWS asegura que las empresas puedan gestionar sus recursos de forma eficiente, ajustándolos a las necesidades del momento sin malgastar inversión en capacidad innecesaria.

1.1 Escalabilidad

La escalabilidad permite empezar con los recursos justos necesarios y ajustar la capacidad según la demanda, utilizando escalado horizontal o vertical. Esto asegura que solo se pague por los recursos utilizados, sin preocuparse por no tener suficiente capacidad para atender a los clientes.

Para realizar este escalado automáticamente, se utiliza Amazon EC2 Auto Scaling, un servicio de AWS que ajusta la capacidad de las instancias de Amazon EC2 en función de la demanda. Esto garantiza que la infraestructura siempre esté preparada para satisfacer las necesidades sin desperdiciar recursos.

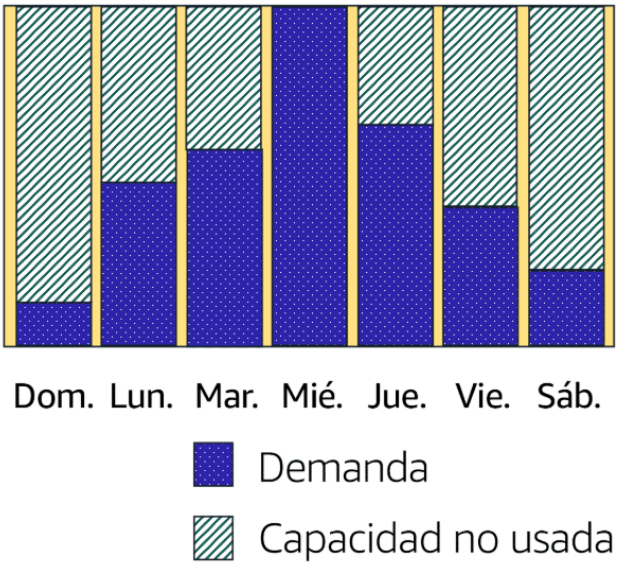
1.2 Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling ajusta automáticamente el número de instancias de Amazon EC2 según la demanda de las aplicaciones. Esto mejora la disponibilidad de las aplicaciones, ya que puedes agregar o eliminar instancias según sea necesario.

Existen dos estrategias en Amazon EC2 Auto Scaling:

1. **Escalado dinámico**: Responde a la demanda cambiante de manera automática.
2. **Escalado predictivo**: Programa el número adecuado de instancias basándose en la demanda anticipada.

Estas estrategias ayudan a gestionar la capacidad de manera eficiente, evitando sobrecargas o recursos sin usar.



Parte 2 (Resumen vídeo)

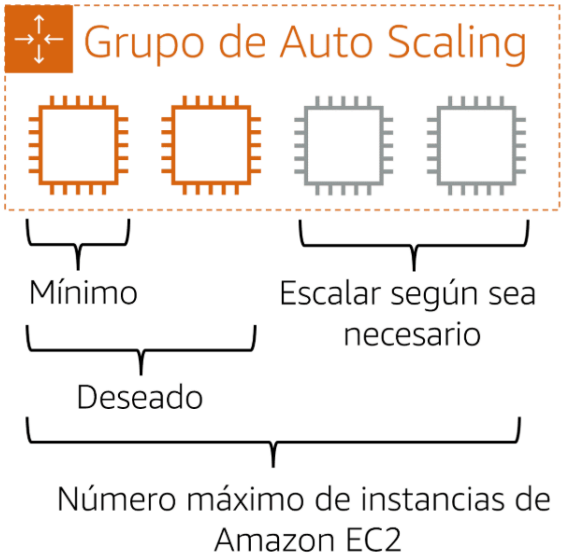
El escalado de demanda en AWS se puede hacer de manera vertical (añadiendo más potencia a las instancias) o horizontal (añadiendo más instancias). El escalado horizontal es más efectivo para gestionar la carga de trabajo, como en el ejemplo de un camarero (Mar) que necesita más clones para atender a más clientes. AWS permite añadir y quitar instancias automáticamente con Amazon EC2 Auto Scaling, ajustando la capacidad según la demanda, lo que optimiza los recursos y satisface tanto a los clientes como a los directores financieros.

Ejemplo: AMAZON EC2 AUTO SCALING

Amazon EC2 Auto Scaling permite gestionar la potencia de computación de manera flexible, añadiendo o eliminando instancias según la demanda. Al crear un grupo de Auto Scaling, puedes configurar:

1. **Capacidad mínima**: el número mínimo de instancias que deben estar siempre activas (por ejemplo, al menos una instancia en ejecución).
2. **Capacidad deseada**: el número de instancias que deseas ejecutar, que puede ser mayor que el mínimo.
3. **Capacidad máxima**: el límite de instancias que puedes tener, por ejemplo, hasta un máximo de cuatro instancias.

Este enfoque asegura que solo pagas por las instancias que usas en cada momento, lo que permite una arquitectura rentable y adaptable que mejora la experiencia del cliente mientras reduce los costos.



Dirección de tráfico con Elastic Load Balancing

Introducción (Resumen vídeo)

Elastic Load Balancing (ELB) distribuye automáticamente el tráfico entre instancias de EC2 para evitar sobrecargas. Escala según la demanda sin intervención manual y mantiene la disponibilidad de los servicios. Además, simplifica la gestión del tráfico en arquitecturas desacopladas, como entre frontend y backend. ELB es esencial para garantizar la eficiencia y alta disponibilidad en AWS.

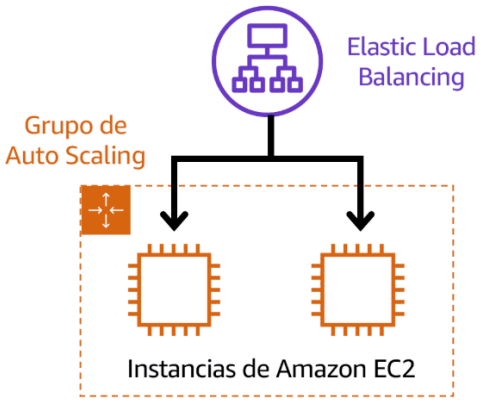
1.1 Elastic Load Balancing

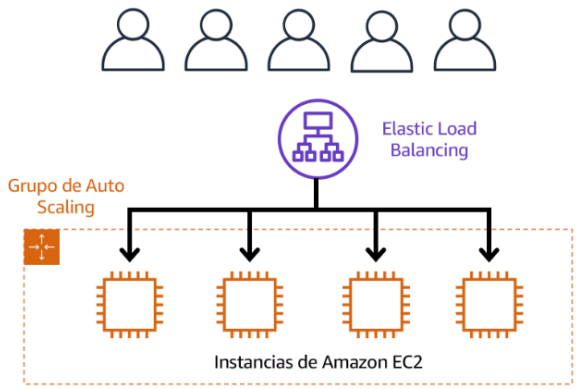
Elastic Load Balancing (ELB) distribuye automáticamente el tráfico entrante entre varios recursos, como instancias EC2. Actúa como un punto único de contacto para todo el tráfico web, dirigiéndolo a las instancias disponibles. ELB trabaja en conjunto con Amazon EC2 Auto Scaling para garantizar el rendimiento y la disponibilidad de las aplicaciones al ajustar dinámicamente los recursos según la demanda.

Ejemplo: Elastic Load Balancing

Periodo de baja demanda:
Cuando la demanda es baja, solo se activan las instancias de EC2 necesarias para atender a los pocos clientes. Es decir, las "cajas registradoras" (instancias de EC2) se abren según la cantidad de clientes que necesiten servicio, evitando recursos innecesarios.

Periodo de alta demanda:
A medida que aumenta la demanda, Elastic Load Balancing escala automáticamente añadiendo más instancias de EC2 y distribuye las solicitudes entre ellas. Esto es similar a un empleado que dirige a los clientes a la caja adecuada, asegurando que los pedidos se manejen de manera equitativa entre las instancias activas.





Mensajería y cola

Introducción (Resumen vídeo)

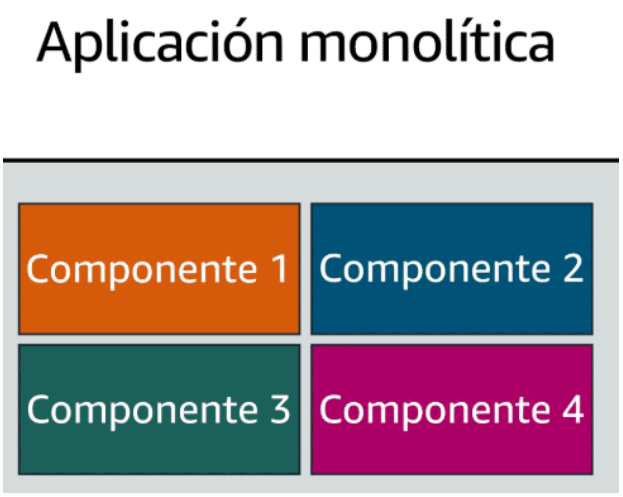
Mensajería y colas:

- **Problema de sincronización:** En un proceso tradicional, como el de un cajero y un barista en una cafetería, si ambos no están sincronizados, pueden ocurrir retrasos y atascos. Si un barista está ocupado o de descanso, el pedido podría quedar en espera hasta que esté disponible, lo que ralentiza el proceso.
- **Solución: Introducción de un buffer o cola:** Para evitar los retrasos, se introduce un buffer, como un tablón de comandas, donde los pedidos se colocan temporalmente. Esto es similar a la mensajería y colas en aplicaciones, donde los mensajes (como los pedidos) se colocan en una cola para ser procesados por otras aplicaciones cuando estén listas.
- **Acoplamiento fuerte vs. acoplamiento débil:**
    - **Acoplamiento fuerte:** Si un componente falla, afecta a todo el sistema (como si un barista no está disponible para recibir un pedido, lo que retrasa el proceso).
    - **Acoplamiento débil:** Si un componente falla, se aísla sin afectar al resto del sistema. Por ejemplo, si la aplicación B falla, la A sigue funcionando y los mensajes siguen esperando en la cola hasta que se procesen.
- **Servicios de AWS:**
    - **Amazon SQS (Simple Queue Service):** Permite enviar, almacenar y recibir mensajes entre componentes de software sin perder mensajes. Se utiliza para colocar los mensajes (pedidos) en una cola hasta que sean procesados. AWS gestiona la infraestructura subyacente, garantizando fiabilidad, escalabilidad y facilidad de uso.
    - **Amazon SNS (Simple Notification Service):** Utiliza el modelo de publicación/suscripción (pub/sub), donde un tema (canal) difunde mensajes a varios suscriptores, como colas de SQS, funciones Lambda o servicios HTTP. SNS también permite enviar notificaciones a usuarios finales mediante SMS, correo electrónico o notificaciones push.

Este enfoque desacoplado mejora la fiabilidad del sistema, permitiendo que las aplicaciones sigan funcionando incluso si algunos componentes fallan, mientras que los mensajes se mantienen en espera para ser procesados.

1.1Aplicaciones monolíticas

- **Arquitectura monolítica:**
  - Una aplicación monolítica consta de varios componentes estrechamente acoplados que se comunican entre sí, como bases de datos, servidores, interfaces de usuario y lógica empresarial.
  - Estos componentes están interconectados y dependen unos de otros para funcionar correctamente. Si uno de los componentes falla, es probable que todo el sistema falle.
- **Problema en aplicaciones monolíticas:**
  - El fallo de un componente afecta a otros, lo que puede llevar a la caída de la aplicación completa. Esto dificulta la escalabilidad y la resiliencia del sistema.



1.2Microservicios

Arquitectura de microservicios:

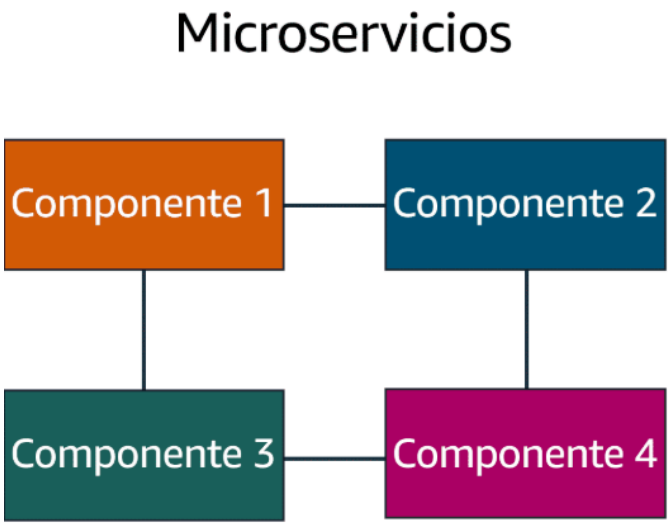
- Los componentes de la aplicación tienen un **acoplamiento débil**, lo que significa que si un componente falla, los demás continúan funcionando sin que la aplicación entera se vea afectada.
- Cada componente tiene una función específica, y están comunicados entre sí de manera independiente.

Ventaja de los microservicios:

- **Resiliencia:** Si un componente falla, los demás siguen operando sin interrupciones.
- Facilita la escalabilidad, el mantenimiento y la implementación de nuevos servicios.

Integración en AWS:

- **Amazon SNS (Simple Notification Service):** Permite la comunicación de eventos o mensajes entre componentes, siguiendo el modelo de **publicación/suscripción**.
- **Amazon SQS (Simple Queue Service):** Permite el envío y almacenamiento de mensajes entre aplicaciones, gestionando la carga y garantizando que no se pierdan mensajes.



1.3Amazon Simple Notification Service (Amazon SNS)

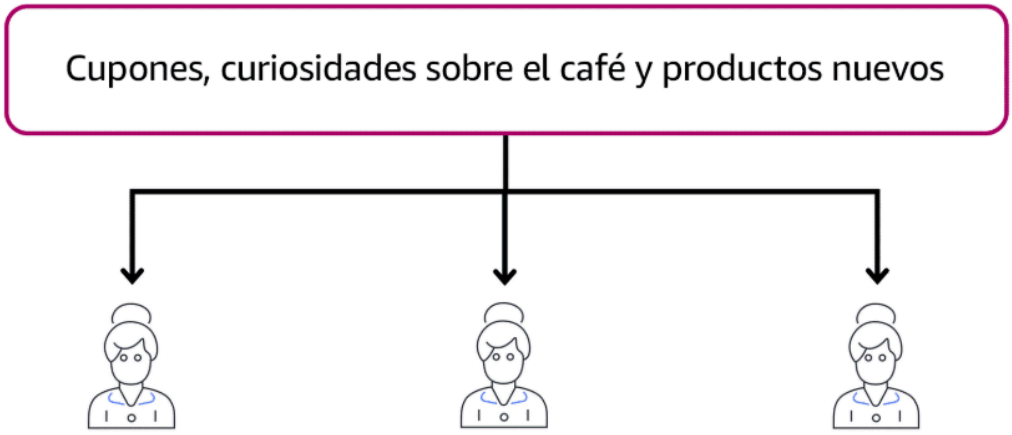
Descripción: Amazon SNS es un servicio de publicación/suscripción. Un editor (publicador) envía mensajes a un tema de SNS, y los suscriptores reciben esos mensajes.
Funcionamiento: Similar a la cafetería, donde el cajero pasa los pedidos al camarero para su preparación. En este caso, el editor "publica" un mensaje y los suscriptores lo reciben.
Tipos de suscriptores:

- Servidores web
- Direcciones de correo electrónico
- Funciones de AWS Lambda
- Otros puntos de enlace

Modelo de comunicación: Los mensajes se envían a un tema y se difunden a todos los suscriptores vinculados a ese tema.

Paso 1: Publicación de actualizaciones de un tema único

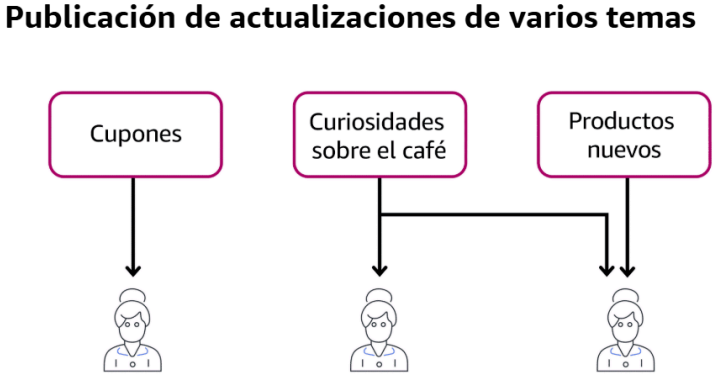
- **Descripción**: Un único boletín informativo agrupa todas las actualizaciones y noticias sobre diversos temas, como cupones, curiosidades sobre el café y productos nuevos.
- **Funcionamiento**: Todos los clientes suscritos al boletín reciben las actualizaciones sobre todos los temas, sin distinción.
- **Cambio de estrategia**: Después de recibir comentarios de algunos clientes que prefieren boletines específicos para cada tema, los propietarios deciden probar la estrategia de ofrecer boletines independientes. Esto permite que los suscriptores reciban solo los boletines que les interesan, en lugar de uno solo con todos los temas agrupados.



Paso 2: Publicación de actualizaciones de varios temas

- **Descripción**: La cafetería divide el boletín único en tres boletines independientes, cada uno dedicado a un tema específico: cupones, curiosidades sobre el café y productos nuevos.
- **Funcionamiento**: Los suscriptores ahora recibirán actualizaciones solo sobre los temas a los que se han suscrito.
- **Ejemplos de suscripciones**:
    - El primer cliente se suscribe solo al tema de cupones.
    - El segundo cliente se suscribe solo al tema de curiosidades sobre el café.
    - El tercer cliente se suscribe a los tres temas: cupones, curiosidades sobre el café y productos nuevos.

Esta estructura permite a los clientes recibir solo la información que les interesa, mejorando la personalización y relevancia de las actualizaciones.



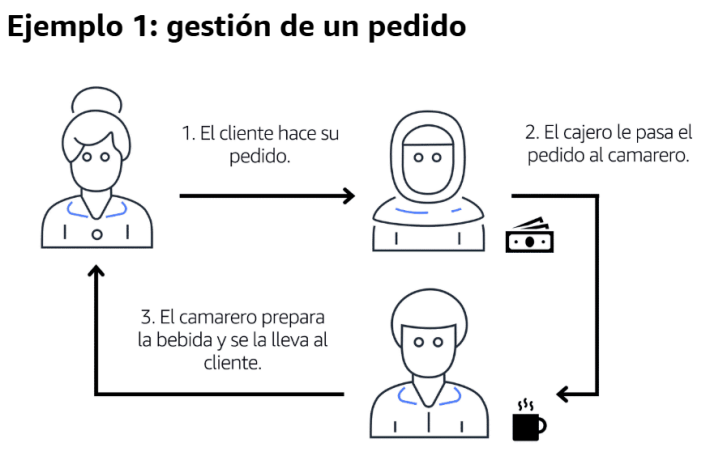
1.4 Amazon Simple Queue Service (Amazon SQS)

- **Descripción**: Amazon SQS es un servicio de cola de mensajes que permite enviar, almacenar y recibir mensajes entre componentes de software de manera confiable.
- **Características**:
    - Los mensajes se envían a una cola.
    - Un usuario o servicio recupera los mensajes de la cola, los procesa y luego los elimina de la cola.
    - Garantiza que no se pierdan mensajes, incluso si otros servicios no están disponibles.
- **Ventajas**:
    - El servicio ayuda a desacoplar los componentes de una aplicación, permitiendo que sigan funcionando de forma independiente y sin interrupciones si un componente falla.
    - Ideal para aplicaciones con alta demanda de procesamiento de mensajes, donde los mensajes se gestionan en una cola para un procesamiento posterior.

Ejemplo 1: Gestión de un pedido con Amazon SQS

- **Contexto**: En una cafetería, el proceso de gestión de pedidos se ve afectado por la falta de coordinación entre el cajero y el camarero. Cuando el cajero toma un pedido y no puede entregárselo al camarero porque este está ocupado o en descanso, se genera un retraso en el proceso, lo que afecta la experiencia del cliente.
- **Solución**: Para solucionar este problema, se introduce una cola, similar a cómo funciona Amazon SQS:
    - El cajero, en lugar de entregar el pedido directamente al camarero, lo coloca en una cola de pedidos.
    - El camarero, cuando está disponible, toma el pedido de la cola, lo prepara y lo entrega al cliente.
- **Beneficio**: La cola permite desacoplar a los dos componentes (el cajero y el camarero), evitando los retrasos causados por la falta de coordinación. Si el camarero está ocupado, el pedido simplemente espera en la cola hasta que el camarero pueda atenderlo, asegurando que los pedidos se gestionen de manera más eficiente sin que se pierdan.

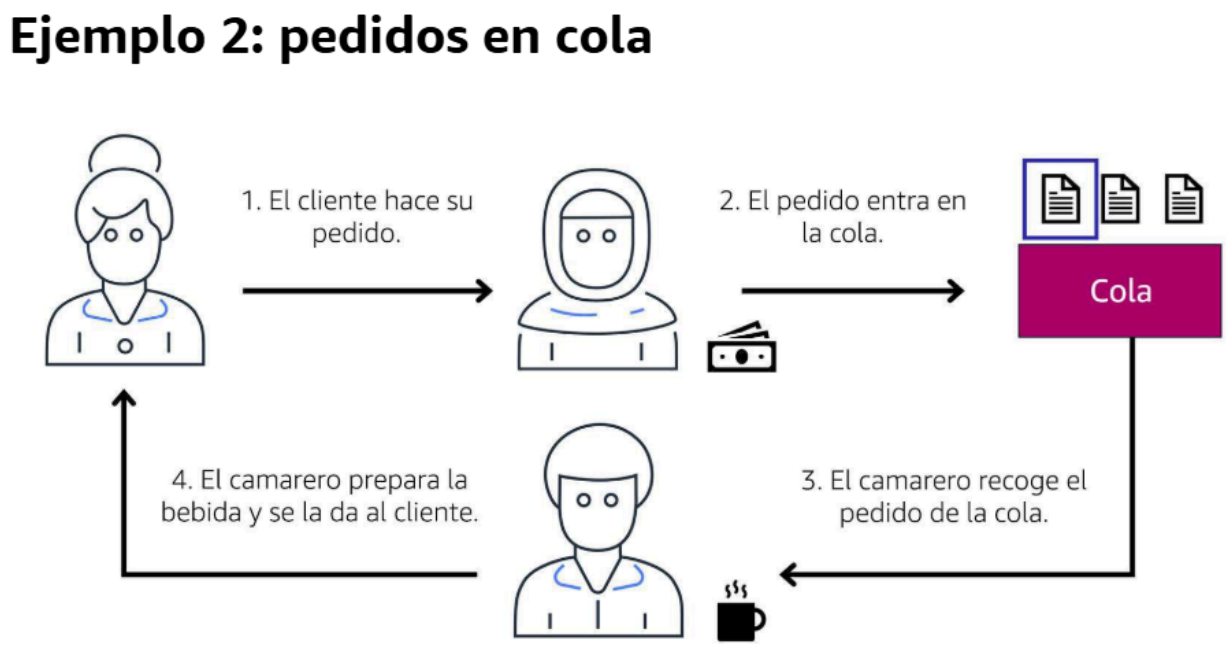
Este ejemplo demuestra cómo una cola de mensajes, como Amazon SQS, puede mejorar la eficiencia y la sincronización en sistemas con múltiples componentes.



Ejemplo 2: Pedidos en cola con Amazon SQS

- **Contexto**: Al igual que en el primer ejemplo, el cajero y el camarero son dos componentes independientes de la aplicación, y Amazon SQS funciona como un servicio de cola de mensajes entre ellos, mejorando la eficiencia del proceso.
- **Proceso**:
    1. **Pedido del cliente**: El cliente realiza un pedido al cajero, quien coloca el pedido en una cola (como un tablón de pedidos o búfer).
    2. **Cajero sigue recibiendo pedidos**: Aunque el camarero esté ocupado o en descanso, el cajero puede seguir aceptando nuevos pedidos y agregarlos a la cola.
    3. **Camarero consulta la cola**: El camarero, cuando está disponible, revisa la cola de pedidos y toma uno para preparar.
    4. **Preparación y entrega**: El camarero prepara la bebida y la entrega al cliente.
    5. **Eliminación del pedido**: Después de entregar la bebida, el camarero elimina el pedido de la cola.
- **Beneficio**: Este enfoque asegura que los pedidos no se pierdan y que el proceso continúe sin que se detenga. La cola actúa como un espacio de almacenamiento intermedio, lo que permite que los componentes estén desacoplados y sigan funcionando de manera eficiente, sin que el cajero dependa de la disponibilidad inmediata del camarero.

Este ejemplo muestra cómo el uso de una cola de mensajes facilita la gestión de procesos asincrónicos en sistemas con componentes desacoplados.



Servicios de computación adicionales

1.1 Computación sin servidor

¿Qué es la computación sin servidor?

- Aunque las aplicaciones siguen ejecutándose en servidores, no es necesario que el usuario los configure ni los administre.
- Permite a los desarrolladores enfocarse en crear productos y nuevas funcionalidades, dejando el mantenimiento de servidores a AWS.

Diferencias con servidores virtuales (Amazon EC2)

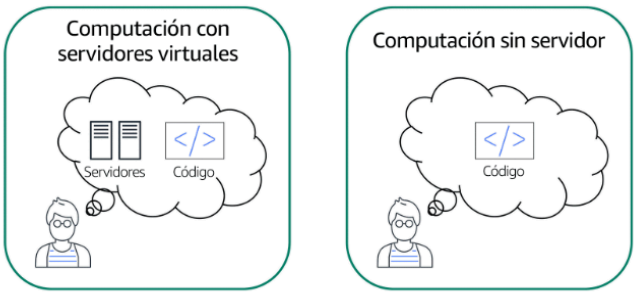
- En EC2, debes:
    - Configurar servidores virtuales (instancias).
    - Subir el código de tu aplicación.
    - Administrar los servidores durante la ejecución.
- Con la computación sin servidor, solo te ocupas de tu código; AWS gestiona la infraestructura.

Ventajas clave

- **Escalabilidad automática**: Las aplicaciones ajustan su capacidad según las necesidades (como memoria y rendimiento).
- **Menor carga de gestión**: No necesitas preocuparte por tareas de mantenimiento.

Ejemplo en AWS

- **AWS Lambda**: Servicio sin servidor que ejecuta tu código automáticamente, manejando el aprovisionamiento, escalado y mantenimiento, ideal para aplicaciones modernas y dinámicas.



Comparación entre la computación con servidores virtuales (teniendo en cuenta servidores y código) y la computación sin servidor (teniendo en cuenta solo el código).

1.2 AWS Lambda

¿Qué es AWS Lambda?
AWS Lambda es un servicio de computación sin servidor que permite ejecutar código sin la necesidad de administrar servidores. Solo pagas por el tiempo de computación que consumes, es decir, únicamente cuando tu código está en ejecución.

Ventajas principales:

- **Costo eficiente**: Paga solo por el tiempo de ejecución del código.
- **Flexibilidad**: Ideal para cualquier tipo de aplicación o backend sin preocuparte por la infraestructura.
- **Sin administración**: AWS gestiona el aprovisionamiento, el escalado y el mantenimiento.

Ejemplo práctico:
Imagina una función que cambia automáticamente el tamaño de imágenes subidas a AWS. Cada vez que se carga una imagen nueva:

1. La función Lambda se activa.
2. Ejecuta el código para redimensionar la imagen.
3. Solo pagas por el tiempo empleado en este proceso.

Cómo funciona AWS Lambda:

1. Subes el código a Lambda.
2. Configuras la función para que se active mediante una fuente de eventos (ejemplo: servicios de AWS, aplicaciones móviles, o puntos de enlace HTTP).
3. Lambda ejecuta el código únicamente cuando se detecta el evento configurado.
4. Pagas solo por los recursos consumidos durante la ejecución.

Este servicio es ideal para tareas automáticas y eventos específicos que no requieren procesos largos.

1.3 Contenedores

AWS permite crear y ejecutar aplicaciones en contenedores, ofreciendo un enfoque estándar para empaquetar el código de la aplicación y sus dependencias en un único objeto. Este enfoque garantiza un entorno seguro, fiable y escalable.

Ejemplo 1: Un host con varios contenedores

Contexto:
Un desarrollador de aplicaciones utiliza un entorno en su equipo que difiere del que usa el personal de operaciones de TI. Para garantizar la coherencia del entorno de la aplicación independientemente del despliegue, emplea una estrategia basada en contenedores. Esto permite:

- Reducir el tiempo dedicado a depurar aplicaciones.
- Diagnosticar más rápidamente las diferencias en los entornos de cómputo.
- Asegurar una experiencia consistente desde el desarrollo hasta la producción.

Un contenedor incluye:

1. La aplicación.
2. Bibliotecas y dependencias necesarias.
3. Un sistema operativo reducido.
4. Un servidor para ejecutar la aplicación.

Ejemplo 2: Decenas de hosts con cientos de contenedores

Contexto: Cuando las necesidades de una organización crecen, es posible que deban gestionar aplicaciones en una infraestructura de mayor escala, como:

- Decenas de hosts con cientos de contenedores.
- Cientos de hosts con miles de contenedores.

Desafíos a gran escala:

- Supervisar el uso de recursos como memoria y CPU.
- Gestionar la seguridad.
- Configurar el registro y monitoreo.

Solución en AWS:
Para abordar estos desafíos, AWS ofrece servicios como:

- **Amazon Elastic Container Service (ECS):** Coordinación y escalabilidad de contenedores.
- **Amazon Elastic Kubernetes Service (EKS):** Gestión de contenedores basada en Kubernetes.

- **AWS Fargate:** Ejecución de contenedores sin necesidad de administrar servidores subyacentes.

Estos servicios permiten escalar las aplicaciones sin aumentar la carga administrativa, ofreciendo una solución eficiente para la gestión de contenedores en diferentes niveles de complejidad.

1.4 Tipos de contenedores

Amazon Elastic Container Service (Amazon ECS):
Es un servicio de administración de contenedores escalable y de alto rendimiento que permite ejecutar y escalar aplicaciones en contenedores en AWS. Soporta contenedores Docker (tanto la versión de código abierto como la versión empresarial) y permite iniciar y detener aplicaciones mediante llamadas a la API.

Amazon Elastic Kubernetes Service (Amazon EKS):
Es un servicio completamente gestionado para ejecutar Kubernetes en AWS. Kubernetes es una plataforma de código abierto que facilita la implementación y gestión de aplicaciones en contenedores a gran escala. AWS colabora activamente en la comunidad de Kubernetes y permite aplicar actualizaciones fácilmente.

AWS Fargate:
Es un motor de computación sin servidor para contenedores que funciona con Amazon ECS y EKS. No requiere la gestión de servidores, ya que AWS Fargate se encarga de la infraestructura. Esto permite centrarse en el desarrollo de aplicaciones, pagando solo por los recursos necesarios para ejecutar los contenedores.

RESUMEN MÓDULO 2

La computación en la nube es la entrega de recursos de TI bajo demanda a través de Internet, con precios de pago por uso. AWS ofrece servicios como **Amazon EC2** para gestionar servidores virtuales, que pueden ser escalados vertical o horizontalmente. **Elastic Load Balancer** distribuye el tráfico entre instancias EC2. AWS también tiene modelos de precios flexibles, como pago bajo demanda y planes de ahorro.

En cuanto a mensajería, AWS proporciona **SQS** para desacoplar sistemas y **SNS** para enviar mensajes a varios suscriptores. Además, AWS ofrece servicios de contenedores como **ECS** y **EKS**, y herramientas sin servidor como **AWS Fargate** y **AWS Lambda** para ejecutar contenedores y código sin gestionar infraestructura.

Módulo 3: Infraestructura global y fiabilidad

Introducción

1.1 Introducción al módulo (resumen vídeo).

El concepto de alta disponibilidad asegura que, incluso ante imprevistos como una cabalgata o un desastre, los servicios sigan funcionando. En el ejemplo, una cadena de cafeterías garantiza que, si un local no puede atender debido a un evento, los clientes puedan ir a otro. AWS aplica este enfoque mediante su infraestructura global, distribuyendo recursos en diferentes **Regiones** para evitar fallos catastróficos en un solo centro de datos. Esto asegura que los servicios continúen funcionando sin interrupciones, garantizando alta disponibilidad y tolerancia a fallos.

La infraestructura global de AWS se asemeja al ejemplo de la cafetería: si algo afecta a un centro de datos en un lugar (como un desfile o una inundación), los usuarios pueden acceder a los servicios desde otro centro cercano. Esta distribución asegura que los servicios sigan disponibles, incluso en caso de fallos o imprevistos.

Infraestructura global de AWS

1.1 Introducción al módulo (resumen vídeo).

La infraestructura global de AWS está diseñada para garantizar alta disponibilidad y seguridad. Similar a una cadena de cafeterías, AWS crea regiones alrededor del mundo, cada una con varios centros de datos, para evitar problemas como desastres naturales o cortes de energía. Estas regiones están aisladas y cumplen con las leyes locales.

Al elegir una región, considera cuatro factores:

1. **Conformidad**: Cumplir con regulaciones locales.
2. **Proximidad**: Cercanía a los clientes para minimizar latencia.
3. **Características disponibles**: Algunas regiones pueden tener servicios específicos.
4. **Precios**: Los costos pueden variar según la ubicación.

1.2 Selección de una región.

Al seleccionar una región para tus servicios en AWS, debes considerar cuatro factores clave:

1. **Conformidad**: Algunas empresas deben cumplir con regulaciones locales sobre la ubicación de los datos. Por ejemplo, si necesitas que los datos permanezcan en el Reino Unido, elegirías la región de Londres.
2. **Proximidad**: Elegir una región cerca de tus clientes reduce la latencia y mejora el rendimiento. Si tu sede está en Washington DC y tus clientes en Singapur, podrías usar la región de Virginia para la infraestructura y la de Singapur para ejecutar las aplicaciones.
3. **Servicios disponibles**: Las regiones pueden no tener todos los servicios de AWS disponibles. Por ejemplo, Amazon Braket (computación cuántica) solo está disponible en algunas regiones, por lo que si tu empresa necesita este servicio, debes elegir una región que lo ofrezca.
4. **Precios**: Los costos pueden variar según la región. Por ejemplo, ejecutar una carga de trabajo en São Paulo podría ser un 50% más caro que en Oregón debido a la estructura fiscal local.

Estos factores deben ser evaluados cuidadosamente para seleccionar la región más adecuada para tu negocio.

1.3 Zonas de disponibilidad

Para garantizar alta disponibilidad y evitar interrupciones debido a desastres, AWS no solo tiene centros de datos en sus regiones, sino que organiza estos centros en "zonas de disponibilidad" (AZ), cada una con múltiples centros de datos independientes. Cada AZ tiene redundancia en alimentación, redes y conectividad.

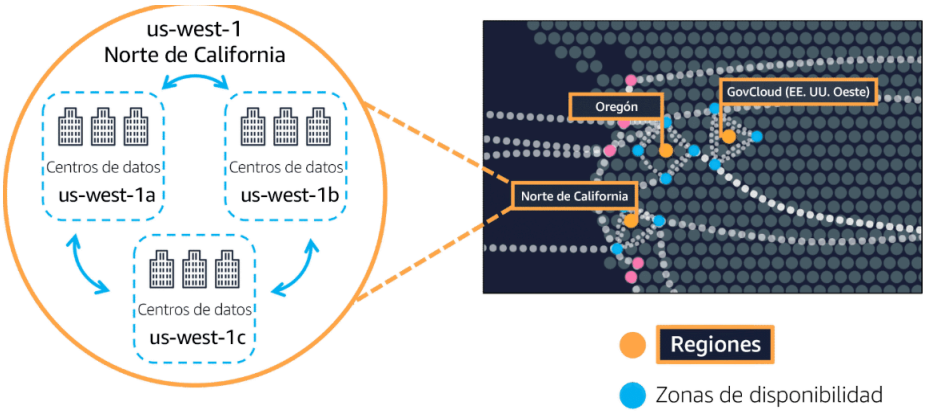
AWS recomienda distribuir tus instancias (como Amazon EC2) en al menos dos AZ dentro de una región para asegurar que, si una zona falla por un desastre, las aplicaciones sigan funcionando sin interrupciones. Las zonas de disponibilidad están separadas físicamente y pueden estar a decenas de kilómetros entre sí, manteniendo una latencia baja.

Además, muchos servicios de AWS operan a nivel regional y son automáticamente altamente disponibles sin costo adicional ni intervención del usuario. Por lo tanto, usar al menos dos zonas de disponibilidad dentro de una región es clave para garantizar la resiliencia de las aplicaciones.

¿Qué es una zona de disponibilidad?

Una zona de disponibilidad (AZ) en AWS es un centro de datos individual o un grupo de centros de datos dentro de una región. Las zonas de disponibilidad están distribuidas a una distancia de decenas de kilómetros entre sí para asegurar que, si ocurre un desastre en una zona, las demás no se vean afectadas. Esto ayuda a mantener la alta disponibilidad y minimizar el impacto en los servicios. Además, la proximidad entre las zonas garantiza baja latencia, lo que significa que los datos viajan rápidamente entre las zonas.

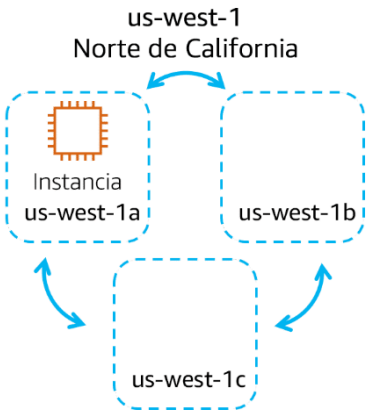
Ejecutar instancias de Amazon EC2 en varias zonas de disponibilidad asegura que, si una zona experimenta un problema, las otras zonas puedan continuar funcionando y ofrecer disponibilidad continua.



Destacados en la región us-west-1. Norte de California, Oregón y GovCloud (EE. UU. Oeste) son regiones independientes. La región Norte de California se denomina us-west-1 y esta región contiene tres zonas de disponibilidad (AZ) (1a, 1b y 1c). En cada zona de disponibilidad (AZ), hay tres centros de datos.

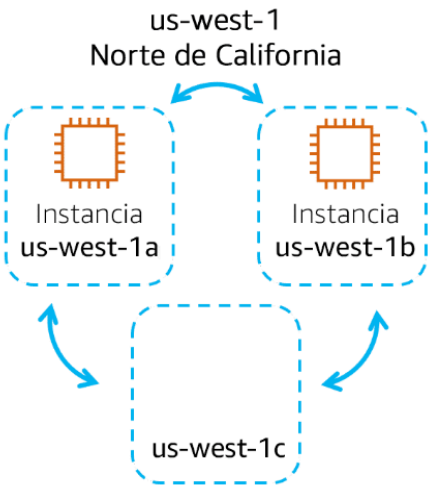
Ejemplo de ejecución de instancias de Amazon EC2 en varias zonas de disponibilidad

Instancia de Amazon EC2 en una única zona de disponibilidad



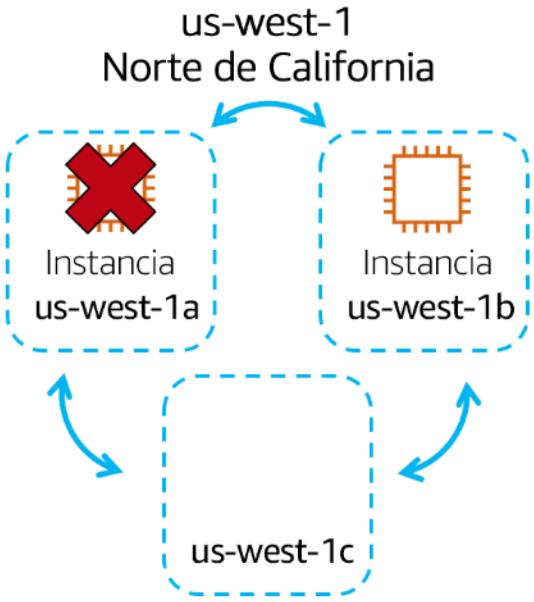
Supongamos que ejecutas una aplicación en una única instancia de Amazon EC2 en la región del Norte de California. La instancia se ejecuta en la zona de disponibilidad us-west-1a. Si us-west-1a fallara, perdería su instancia.

Instancias de Amazon EC2 en varias zonas de disponibilidad



Una práctica recomendada es ejecutar aplicaciones en, al menos, dos zonas de disponibilidad de una región. En este ejemplo, puedes elegir ejecutar una segunda instancia de Amazon EC2 en us-west-1b.

Error en la zona de disponibilidad



Si us-west-1a fallase, tu aplicación seguiría ejecutándose en us-west-1b.

Ubicaciones periféricas

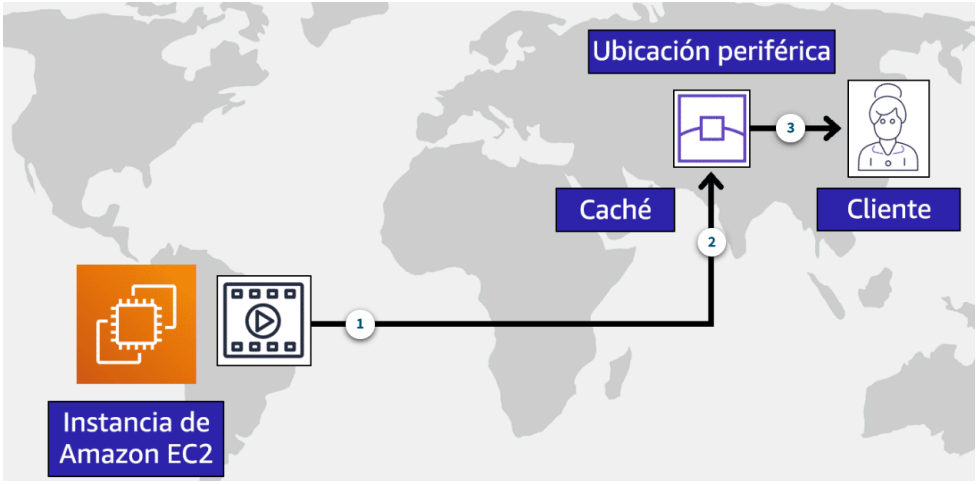
1.1 Intro (resumen vídeo).

AWS ofrece una infraestructura global diseñada para mejorar la experiencia de los clientes. Si tienes clientes distribuidos globalmente, puedes usar redes de entrega de contenido (CDN) como Amazon CloudFront para almacenar datos cerca de ellos, reduciendo la latencia. Amazon CloudFront entrega contenido con alta velocidad y baja latencia, utilizando ubicaciones periféricas en todo el mundo. Además, Amazon Route 53 se encarga de dirigir a los clientes a las ubicaciones correctas.

Para empresas que necesitan soluciones en sus instalaciones, AWS ofrece Outposts, una mini-región completamente operativa en su centro de datos. En resumen, las regiones de AWS son áreas geográficas aisladas que contienen zonas de disponibilidad separadas físicamente para ofrecer alta disponibilidad, recuperación ante desastres y baja latencia para los clientes.

1.2 Ubicaciones periféricas.

Una **ubicación periférica** es un sitio que Amazon CloudFront usa para almacenar en la memoria caché copias de tu contenido más cerca de tus clientes para una entrega más rápida.



1.Origen
Supongamos que los datos de tu empresa se almacenan en Brasil y que tienes clientes que viven en China. Para proporcionar contenido a estos clientes, no es necesario trasladar todo el contenido a una de las regiones chinas.

2.Ubicación periférica

En lugar de que los clientes tengan que recibir los datos desde Brasil, puedes almacenar en caché una copia de forma local en una ubicación periférica cercana a tus clientes de China.

3.Cliente
Cuando un cliente de China solicita uno de tus archivos, Amazon CloudFront recupera el archivo de la caché de la ubicación periférica y lo entrega al cliente. El archivo se entrega al cliente más rápido porque proviene de la ubicación periférica cercana a China y no de la fuente original de Brasil.

Cómo aprovisionar recursos de AWS

Parte 1 (resumen vídeo)

En AWS, la interacción con los servicios se realiza a través de API (interfaces de programación de aplicaciones), lo que permite aprovisionar, configurar y gestionar recursos de AWS. Puedes utilizar herramientas como la consola de administración de AWS, la interfaz de línea de comandos (CLI) y los kits de desarrollo de software (SDK) para hacer solicitudes a estas API.

La consola de administración es ideal para comenzar, administrar recursos de forma visual y realizar tareas no técnicas, pero no es recomendada para entornos de producción, ya que aumentarían los errores manuales. La CLI permite automatizar procesos mediante scripts, lo que hace que el aprovisionamiento sea más eficiente y menos propenso a errores. Los SDK facilitan la interacción con AWS utilizando lenguajes de programación, permitiendo a los desarrolladores crear aplicaciones sin tener que interactuar directamente con las API.

1.1 Formas de interactuar con los servidores de AWS

1. Consola de administración de AWS:

La consola de administración de AWS es una interfaz web para acceder y gestionar servicios de AWS. Permite buscar servicios y acceder rápidamente a los más utilizados, además de incluir asistentes y flujos automatizados para facilitar tareas. También está disponible la aplicación móvil de AWS Console, que permite supervisar recursos, ver alarmas y acceder a la facturación, y admite múltiples identidades activas al mismo tiempo.

2. AWS Command Line Interface (AWS CLI):

AWS Command Line Interface (AWS CLI) permite controlar servicios de AWS desde la línea de comandos en Windows, macOS y Linux. Facilita la automatización de acciones a través de scripts, como iniciar instancias de Amazon EC2 o conectarlas a grupos de Auto Scaling, lo que ahorra tiempo y mejora la eficiencia.

3.kits de desarrollo de software (SDK):

Los kits de desarrollo de software (SDK) permiten acceder a los servicios de AWS a través de APIs específicas para tu lenguaje de programación o plataforma. Facilitan la integración de servicios de AWS en aplicaciones existentes o la creación de nuevas aplicaciones en AWS. AWS proporciona documentación y ejemplos de código para lenguajes como C++, Java, .NET, entre otros.

Parte 2 (resumen vídeo)

AWS ofrece varias formas de interactuar con sus servicios, como la consola de administración, la CLI y los SDK. Para automatizar y administrar entornos de AWS, también existen herramientas como AWS Elastic Beanstalk y AWS CloudFormation. Elastic Beanstalk simplifica el aprovisionamiento de entornos basados en EC2, permitiéndote centrarte en las aplicaciones, mientras que CloudFormation utiliza plantillas en JSON o YAML para definir y aprovisionar recursos de manera automatizada y repetible. Estas herramientas permiten crear y gestionar recursos en AWS de manera eficiente, especialmente en entornos complejos.

1.1 AWS Elastic Beanstalk

AWS Elastic Beanstalk permite implementar aplicaciones proporcionando código y configuraciones, y se encarga de gestionar tareas como el ajuste de capacidad, equilibrio de carga, escalado automático y el seguimiento del estado de las aplicaciones.

1.2 AWS CloudFormation

Por otro lado, AWS CloudFormation permite gestionar la infraestructura como código. Con CloudFormation, puedes crear y aprovisionar recursos mediante plantillas, lo que hace que el proceso sea seguro, repetible y automatizado, reduciendo la necesidad de intervención manual. Además, CloudFormation gestiona y revierte cambios automáticamente si detecta errores en la infraestructura.

RESUMEN MÓDULO 3

AWS ofrece infraestructura global distribuida en regiones y zonas de disponibilidad. Se recomienda usar al menos dos zonas para mayor disponibilidad. Existen soluciones como AWS Outposts para usar infraestructura de AWS en tus centros de datos. Para aprovisionar recursos, puedes usar la consola de administración, CLI, SDKs, Elastic Beanstalk y CloudFormation, facilitando la gestión de recursos de forma manual o automatizada.

Módulo 4: Redes

Introducción

1.1 Introducción al módulo (resumen vídeo)

Amazon Virtual Private Cloud (VPC) permite crear una red aislada dentro de AWS, donde puedes lanzar recursos con acceso público o privado. Los recursos públicos, como los cajeros en una cafetería, se colocan en una subred pública para interactuar con los clientes (Internet), mientras que los recursos privados, como los baristas, se ubican en una subred privada para trabajar sin interactuar directamente con los clientes.

Conectividad a AWS

1.1 Introducción (resumen vídeo)

Una VPC (nube virtual privada) en AWS es una red privada donde puedes asignar recursos como instancias EC2 y balanceadores de carga. Los recursos se agrupan en subredes, que controlan si están disponibles públicamente o de forma privada. Para que los recursos públicos reciban tráfico de Internet, se utiliza una puerta de enlace de Internet (IGW), mientras que para recursos privados se utiliza una puerta de enlace privada virtual (VGW), permitiendo conexiones VPN desde redes privadas.

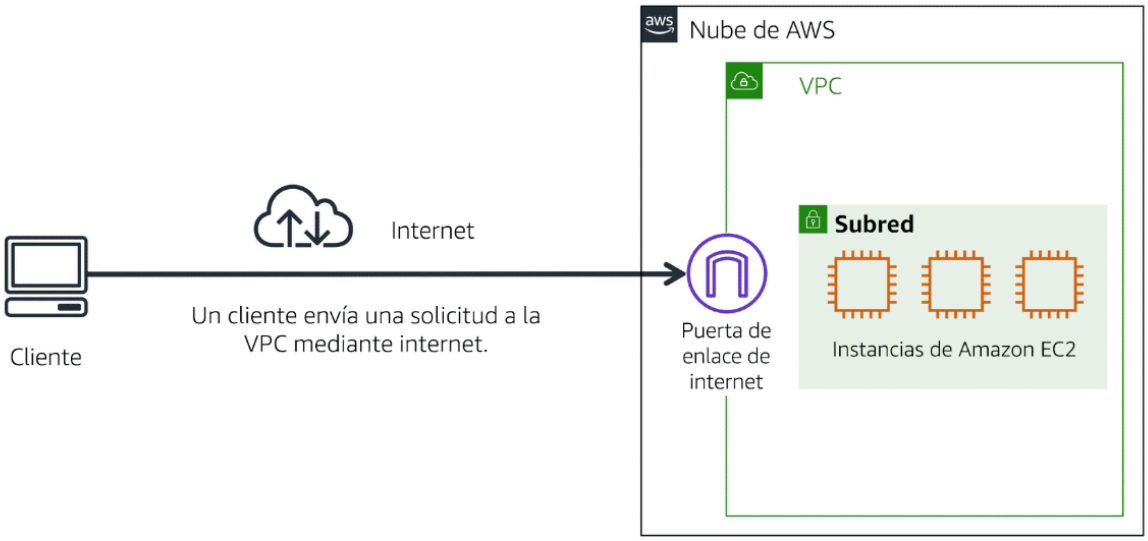
AWS Direct Connect ofrece una conexión privada y exclusiva entre tu infraestructura y AWS, evitando problemas de latencia y ancho de banda compartido que ocurre con las conexiones VPN convencionales. Una VPC puede tener múltiples puertas de enlace para gestionar diferentes tipos de recursos en distintas subredes.

1.2 Amazon Virtual Private Cloud (Amazon VPC)

Amazon Virtual Private Cloud (Amazon VPC) permite crear una sección aislada dentro de AWS Cloud donde se pueden iniciar recursos en una red virtual definida. Esta red virtual está organizada en subredes, que son secciones dentro de la VPC que pueden contener recursos como las instancias de Amazon EC2. VPC ayuda a establecer límites en torno a los recursos y controlar el tráfico de red entre ellos, proporcionando una estructura más segura y organizada para los servicios de AWS.

1.3 Puerta de enlace de internet

Para permitir que el tráfico público de Internet acceda a tu VPC, adjuntas una puerta de enlace de internet a la VPC.

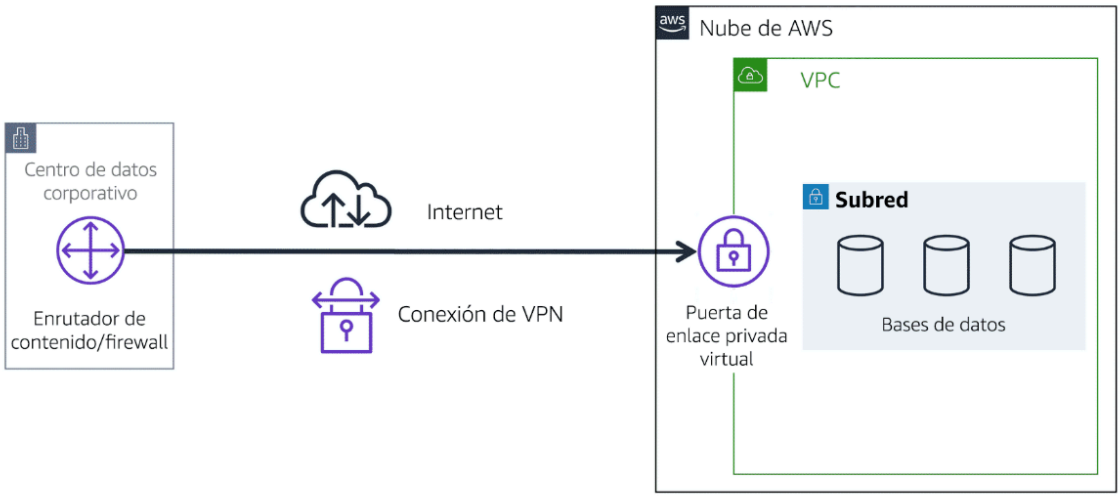


Una puerta de enlace de internet es una conexión entre una VPC e Internet. Podemos imaginar que es como la puerta que utilizan los clientes para entrar a la cafetería. Sin una puerta de enlace de Internet, nadie puede acceder a los recursos de tu VPC.

1.4 Puerta de enlace privada virtual

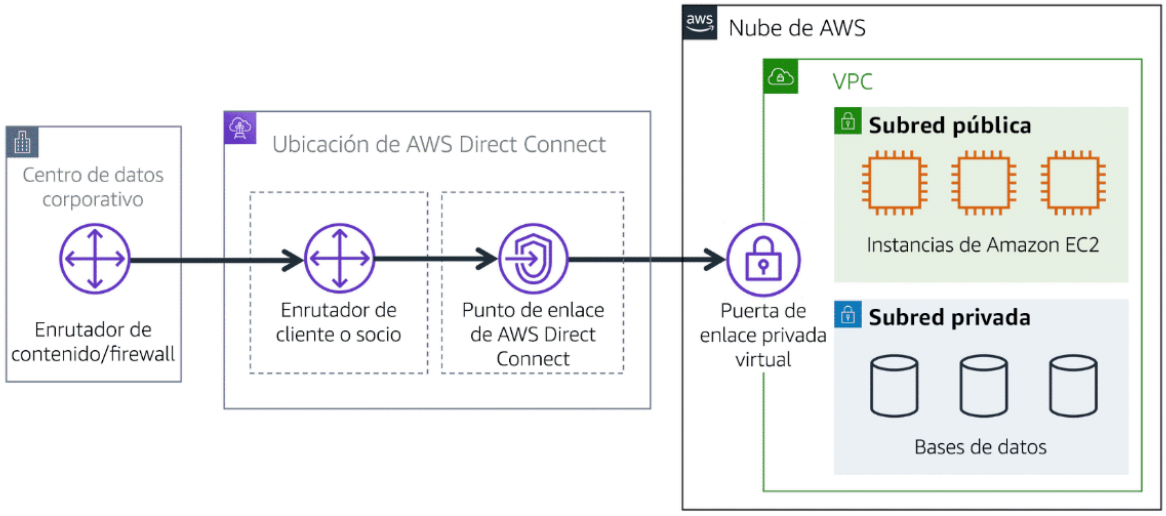
Una puerta de enlace privada virtual permite establecer una conexión de red privada virtual (VPN) entre la VPC y una red privada, como un centro de datos en las instalaciones o una red corporativa interna. Esta puerta de enlace solo permite el tráfico en la VPC si proviene de una red aprobada, garantizando un acceso seguro y restringido a los recursos internos de la VPC. Además, el tráfico está cifrado, proporcionando una capa adicional de protección.

Para entender cómo funciona, imagina que Internet es como el camino entre tu casa y la cafetería. Aunque viajas por la misma carretera que otros clientes, un guardaespaldas te acompaña para protegerte. Este guardaespaldas sería como la conexión VPN, que cifra (o protege) tu tráfico de Internet de las demás solicitudes que lo rodean. Aunque el camino tiene protección adicional, es posible que se produzcan atascos porque se comparte con otros usuarios.



1.5 AWS Direct Connect

AWS Direct Connect es un servicio que establece una conexión privada y dedicada entre un centro de datos y una VPC. Es como un pasillo exclusivo que conecta un edificio de apartamentos con una cafetería, accesible solo para los residentes, lo que permite un acceso directo y privado sin utilizar la vía pública compartida con otros clientes.



La conexión privada que proporciona AWS Direct Connect ayuda a reducir los costes de red y aumentar la cantidad de ancho de banda que puede viajar a través de tu red.

Subredes y listas de control de acceso a la red

1.1 Resumen vídeo

La VPC (Virtual Private Cloud) de AWS actúa como una fortaleza donde el acceso está estrictamente controlado. Las subredes en una VPC permiten segmentar los recursos, y su principal propósito es gestionar el acceso a puertas de enlace. Las subredes públicas tienen acceso a la puerta de enlace de Internet, mientras que las privadas no. Además, el tráfico que entra y sale de las subredes es verificado por las ACL de red (listas de control de acceso), que funcionan como agentes de control de pasaportes, permitiendo solo el tráfico autorizado.

Las ACL de red, a pesar de ser eficaces en la entrada y salida de tráfico de la subred, no pueden verificar el tráfico hacia instancias específicas. Para eso se usan los grupos de seguridad, que funcionan como porteros, controlando el acceso a cada instancia EC2. A diferencia de las ACL, los grupos de seguridad tienen "estado", lo que significa que recuerdan las conexiones permitidas, mientras que las ACL no tienen estado y verifican cada paquete individualmente.

El proceso de envío de paquetes entre instancias en subredes diferentes incluye varias verificaciones en diferentes puntos de control. Un paquete debe pasar por el grupo de seguridad de la instancia de origen, la **NACL** de la subred de origen, la NACL de la subred de destino, y finalmente, el grupo de seguridad de la instancia de destino. Cuando el tráfico regresa, los grupos de seguridad permiten el tráfico de retorno automáticamente, mientras que las NACL verifican cada paquete sin tener en cuenta el estado.

Este sistema de seguridad con grupos de seguridad y NACL asegura una protección robusta y completa de las redes dentro de AWS, aprovechando ambas herramientas para reforzar la seguridad en las arquitecturas modernas.

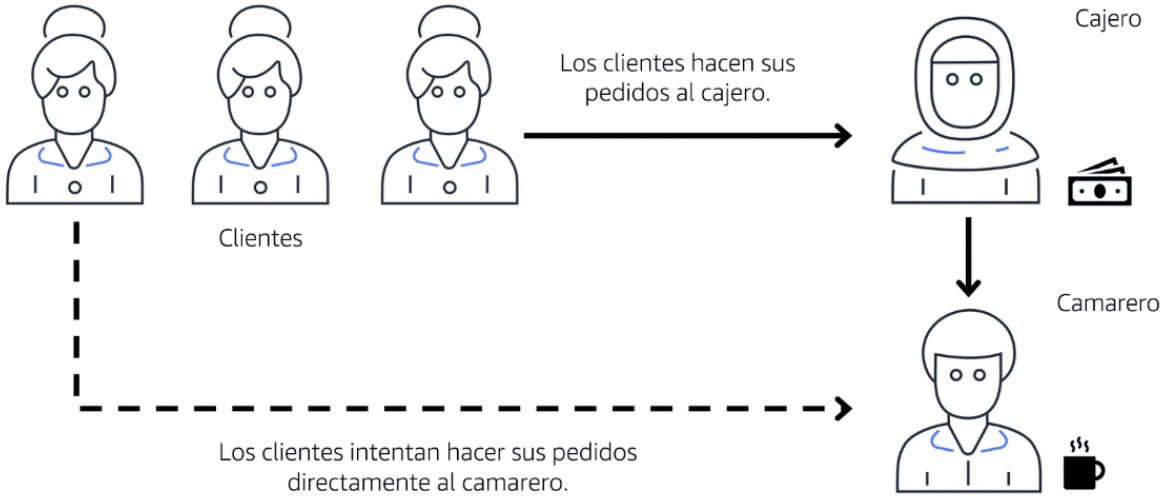
EJEMPLO:

Este ejemplo de la cafetería ilustra cómo las subredes dentro de una VPC en AWS pueden ayudar a controlar y gestionar el tráfico de manera eficiente.

Imagina que la cafetería es como una VPC. El cajero y el camarero representan diferentes puntos de acceso en la red. El cajero (el punto de entrada público) recibe pedidos de los clientes (el tráfico externo) y los pasa al camarero (el punto de salida privado) que se encarga de procesarlos.

Si los clientes intentaran saltarse la cola y pasar directamente al camarero, esto interrumpiría el flujo de trabajo y permitiría el acceso no autorizado a una zona privada. Para evitarlo, los propietarios de la cafetería dividen la zona en dos áreas: el puesto de trabajo del cajero (accesible a los clientes) y el

del camarero (privado). De esta forma, solo el cajero tiene acceso directo a la zona privada del camarero.

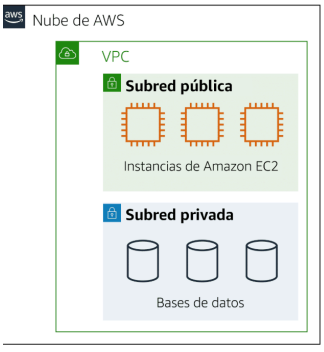


Este proceso es análogo a cómo las subredes en una VPC pueden aislar recursos y controlar cómo fluye el tráfico. Las subredes públicas se asemejan al área del mostrador donde los clientes pueden interactuar directamente, mientras que las subredes privadas son como la zona restringida donde solo ciertos recursos (como los servidores o bases de datos) pueden comunicarse entre sí, sin acceso directo desde fuera de la VPC.

1.2 Subredes

Una **subred** es una división dentro de una VPC que agrupa recursos según sus necesidades operativas o de seguridad. Pueden ser **públicas** o **privadas**:

- **Subredes públicas**: Contienen recursos a los que debe acceder el público, como un sitio web de una tienda en línea. Estos recursos pueden ser accesibles desde fuera de la VPC.
- **Subredes privadas**: Contienen recursos que solo deben ser accesibles a través de la red privada, como bases de datos con información sensible de clientes. Estos recursos no son accesibles directamente desde fuera de la VPC.



Dentro de una VPC, las subredes pueden comunicarse entre sí. Por ejemplo, una aplicación con instancias de Amazon EC2 en una subred pública podría interactuar con bases de datos ubicadas en una subred privada, garantizando la separación y seguridad de los datos.

1.3 Trafico de una red en una VPC

El **tráfico de red en una VPC** se maneja mediante el envío de paquetes de datos. Cuando un cliente solicita datos de una aplicación alojada en la nube de AWS, esa solicitud se envía como un paquete, que es una unidad de datos que circula por la red.

Este paquete entra en la VPC a través de una **puerta de enlace de internet**. Antes de que un paquete pueda entrar o salir de una subred dentro de la VPC, se comprueban los **permisos**. Estos permisos determinan quién ha enviado el paquete y cómo está intentando interactuar con los recursos de la subred.

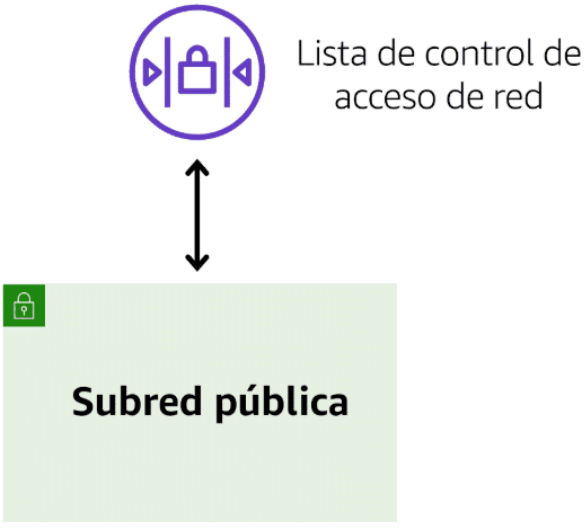
El componente responsable de verificar estos permisos en las subredes de la VPC es la **lista de control de acceso a la red (ACL)**.

1.4 ACL de red

Una ACL (Lista de Control de Acceso a la Red) es un firewall virtual que controla el tráfico entrante y saliente a nivel de subred en una VPC.

Imagina que sales de una cafetería y llegas a un aeropuerto. Los viajeros son como paquetes, y el agente de control de pasaportes actúa como una ACL de red, comprobando las credenciales de los viajeros (paquetes) cuando intentan entrar o salir del país. Si el viajero (paquete) está en la lista aprobada, puede pasar; si no está autorizado o está en la lista de prohibidos, no puede entrar.

En AWS, cada cuenta tiene una ACL de red predeterminada que permite todo el tráfico entrante y saliente. Sin embargo, puedes modificarla añadiendo reglas personalizadas. En el caso de ACL personalizadas, el tráfico entrante y saliente está denegado por defecto hasta que añades reglas para permitirlo. Además, todas las ACL de red tienen una regla de denegación explícita que bloquea cualquier paquete que no coincida con ninguna regla de la lista.

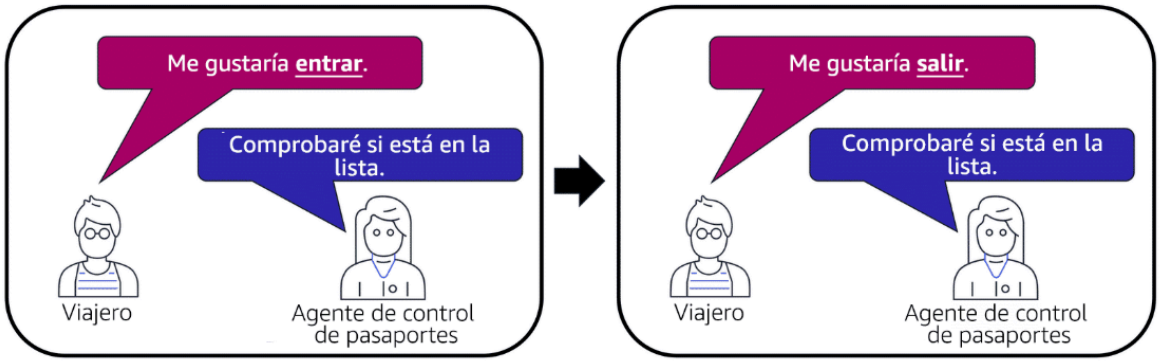


1.5 Filtrado de paquetes sin estado

El **filtrado de paquetes sin estado** de las **ACL de red** significa que no mantienen memoria de los paquetes anteriores. Cada vez que un paquete cruza el borde de la subred, ya sea entrando o saliendo, la ACL realiza una comprobación independiente.

Siguiendo con el ejemplo del viajero que quiere entrar a otro país, cuando un viajero (paquete) entra, el funcionario (ACL) verifica si está autorizado. Sin embargo, si el viajero regresa (como una respuesta de paquete), la ACL no recuerda la entrada anterior y vuelve a comprobar la solicitud según sus reglas.

Una vez que un paquete ingresa a la subred, sus permisos para interactuar con los recursos, como las **instancias de Amazon EC2**, son evaluados por un **grupo de seguridad**. El grupo de seguridad actúa como una segunda capa de control para verificar si el paquete tiene permitido acceder a una instancia específica dentro de la subred.



1.6 Grupos de seguridad

Un **grupo de seguridad** es un firewall virtual que regula el tráfico entrante y saliente de una instancia de **Amazon EC2**. De forma predeterminada, un grupo de seguridad bloquea todo el tráfico entrante y permite todo el tráfico saliente. Se pueden agregar reglas personalizadas para especificar qué tráfico se permite, y cualquier tráfico no autorizado será bloqueado.

Usando la metáfora de un edificio, el **portero** representa el grupo de seguridad. El portero revisa una lista para autorizar la entrada de los invitados (paquetes), pero no hace lo mismo cuando estos salen del edificio. Este comportamiento refleja que un grupo de seguridad permite automáticamente el tráfico de salida, pero solo verifica el tráfico de entrada.

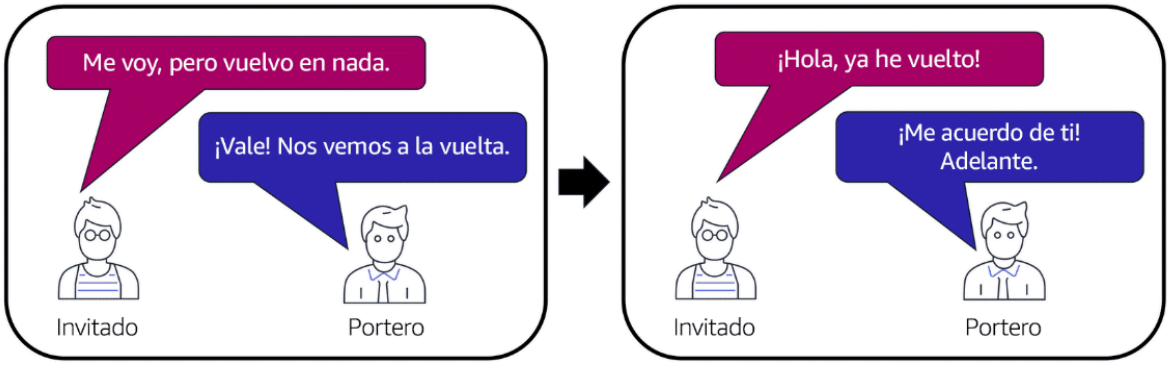
Si tienes varias instancias de EC2 dentro de una subred, puedes asociarlas a un mismo grupo de seguridad o utilizar grupos de seguridad diferentes para cada instancia, según las necesidades de seguridad.

1.7 Filtrado de paquetes con estado

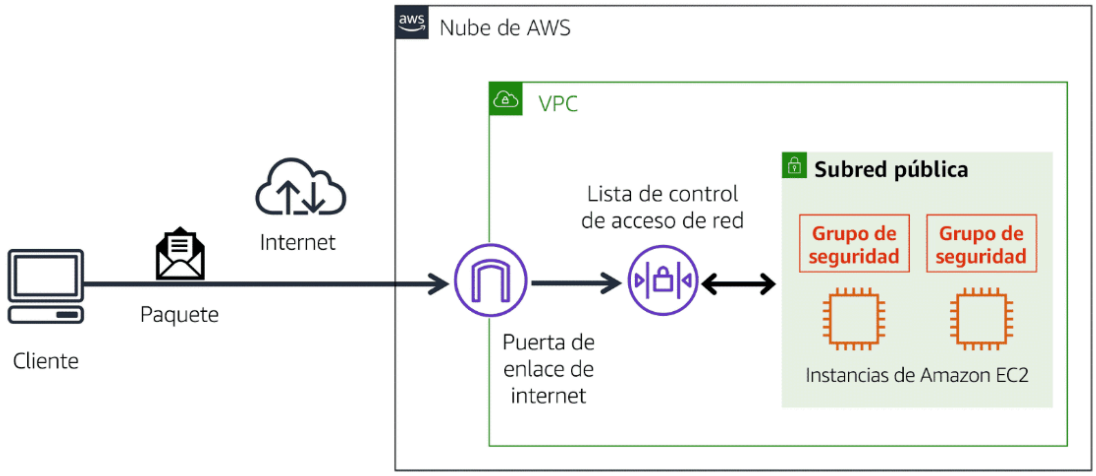
El filtrado de paquetes con estado realizado por los grupos de seguridad permite recordar las decisiones tomadas sobre el tráfico anterior. A diferencia de las ACL de red, que son sin estado, los grupos de seguridad mantienen un registro de las conexiones establecidas, lo que les permite gestionar de manera más eficiente el tráfico de respuesta.

Siguiendo el ejemplo de una solicitud enviada desde una instancia de Amazon EC2 hacia Internet, cuando la respuesta de esa solicitud regresa, el grupo de seguridad "recuerda" que la solicitud original fue aprobada, por lo que no necesita realizar una verificación adicional sobre la respuesta entrante. Esto se debe a que el grupo de seguridad permite el tráfico de regreso sin necesidad de evaluarlo nuevamente, ya que tiene en cuenta el estado de la conexión.

En la analogía del portero, si un invitado fue aprobado para entrar, el portero (grupo de seguridad) no necesitará revisar su autorización cuando decida salir, lo que facilita el flujo de tráfico en ambos sentidos.



Así que, mientras que las ACL de red controlan el tráfico de entrada y salida de una subred sin recordar las decisiones anteriores, los grupos de seguridad manejan el tráfico a nivel de instancia y recuerdan las conexiones establecidas, lo que hace más eficiente y seguro el manejo de las comunicaciones.



Un paquete viaja por Internet desde un cliente hasta la pasarela de Internet y entra en la VPC. A continuación, el paquete pasa por la lista de control de acceso a la red y accede a la subred pública, donde se encuentran dos instancias de EC2.

1.8 Recuperación del componente de VPC

Recupera la finalidad de los siguientes componentes de VPC. Compara tu respuesta eligiendo cada tarjeta de componente de VPC.

Subred privada: Aislar las bases de datos que contienen información personal de los clientes.

Puerta de enlace privada virtual: Crear una conexión de VPN entre la VPC y la red corporativa interna.

Subred pública: Ofrecer soporte al sitio web orientado al cliente.

AWS Direct Connect: Establecer una conexión dedicada entre el centro de datos en las instalaciones y la VPC.

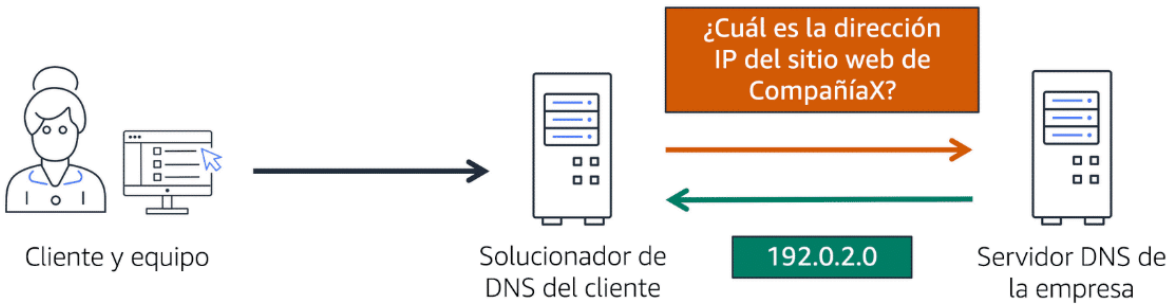
Redes globales

1.1 Sistema de nombres de dominio

El **Sistema de Nombres de Dominio (DNS)** es un servicio que traduce nombres de dominio, como "www.anycompany.com", en direcciones IP, como "192.0.2.0", que los equipos pueden entender. Este proceso se denomina **resolución DNS** y funciona de la siguiente manera:

1. **Solicitud inicial**:
   El cliente introduce el nombre de dominio en su navegador, y esta solicitud se envía al solucionador DNS del cliente.
2. **Consulta al servidor DNS**:
   El solucionador reenvía la solicitud al servidor DNS correspondiente de la empresa para obtener la dirección IP asociada al dominio.
3. **Respuesta del servidor DNS**:
   El servidor DNS responde proporcionando la dirección IP que corresponde al dominio solicitado.

Finalmente, el navegador utiliza esta dirección IP para conectarse al servidor que aloja el sitio web, permitiendo que el cliente acceda al contenido deseado. **DNS actúa como la guía telefónica de Internet**, facilitando la comunicación entre nombres legibles para los humanos y direcciones IP comprensibles para los equipos.



1.2 Amazon Route 53

Amazon Route 53 es un servicio DNS (Sistema de Nombres de Dominio) que permite enrutar de manera confiable las solicitudes de los usuarios a aplicaciones de Internet alojadas en AWS o fuera de ella. Sus principales características incluyen:

- **Enrutamiento de tráfico**: Conecta a los usuarios con recursos como instancias de Amazon EC2 y balanceadores de carga.
- **Gestión de registros DNS**: Permite registrar nuevos nombres de dominio, transferir registros existentes de otros registradores, y gestionar todos los dominios desde una ubicación centralizada.

Integración de Route 53 con Amazon CloudFront

Route 53 colabora con Amazon CloudFront, un servicio de entrega de contenido, para mejorar el rendimiento al entregar contenido desde instancias de Amazon EC2.

Ejemplo: Entrega de contenido con Route 53 y CloudFront

1. **Solicitud inicial**:
   Un cliente accede al sitio web de AnyCompany.
2. **Resolución DNS**:
   Route 53 traduce el dominio del sitio (AnyCompany.com) en su dirección IP, por ejemplo, "192.0.2.0", y envía esta información al cliente.
3. **Entrega mediante CloudFront**:
   La solicitud del cliente se redirige a la ubicación periférica más cercana a través de Amazon CloudFront.
4. **Procesamiento en EC2**:
   CloudFront envía la solicitud al Application Load Balancer, que distribuye el tráfico a una instancia de Amazon EC2 en el grupo de Auto Scaling.

Esto asegura un acceso rápido y confiable al contenido al aprovechar las ubicaciones periféricas de CloudFront y la capacidad de enrutamiento inteligente de Route 53.

RESUMEN MÓDULO 4

El resumen del tema 4 de AWS aborda cómo se pueden simplificar y abstraer las redes en AWS, permitiendo que se gestionen de manera más sencilla, incluso sin tener conocimientos avanzados de topología de redes. A través de conceptos como la VPC (nube virtual privada), las puertas de enlace, las ACL de red y los grupos de seguridad, se facilita la creación de redes seguras que permiten el acceso adecuado y bloquean posibles amenazas.

Además, el tema cubre la conexión a AWS mediante VPN o Direct Connect, asegurando la comunicación segura. También se exploran las redes globales de AWS, como el uso de Route 53 para DNS y CloudFront para distribuir contenido cerca de los usuarios. Aunque este es un resumen general, proporciona una base sólida para comenzar a trabajar con las redes en AWS.

Módulo 5: Almacenamiento y bases de datos

Introducción

1.1 Introducción (resumen vídeo)

El video plantea cómo una cafetería busca implementar un programa de fidelización digital para recompensar a sus clientes y entender mejor sus hábitos de consumo. Esto requiere bases de datos específicas y almacenamiento adecuado. AWS ofrece herramientas para gestionar diferentes tipos de datos y diseñar soluciones adaptadas a estos objetivos.

Almacenes de instancias y Amazon Elastic Block Store (Amazon EBS)

1.1 Introducción (resumen vídeo)

El almacenamiento en Amazon EC2 se basa en almacenamiento en bloques, que organiza datos en bloques individuales para un acceso eficiente, ideal para aplicaciones como bases de datos y sistemas de archivos. Las instancias EC2 pueden usar volúmenes locales llamados **almacén de instancias**, que son temporales y se pierden al detener la instancia.

Para datos persistentes, Amazon ofrece **Elastic Block Store (EBS)**, volúmenes virtuales independientes del hardware subyacente de EC2. EBS permite conservar datos entre paradas y

reinicios de instancias. Los volúmenes de EBS admiten configuraciones personalizables y respaldos incrementales mediante "instantáneas" para garantizar la recuperación y seguridad de los datos en caso de fallos.

1.2 Almacenes de instancias y Amazon EBS

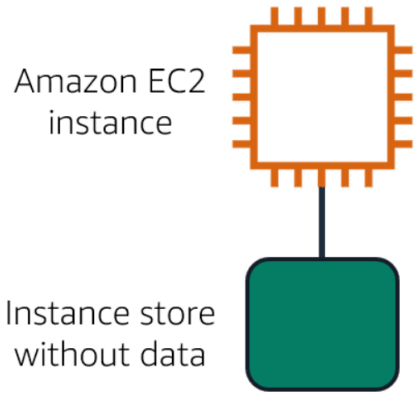
Almacenes de instancias y Amazon EBS: Resumen

1. **Almacenes de instancias**:
    - Proporcionan almacenamiento temporal en bloque para instancias de Amazon EC2.
    - Conectados físicamente al equipo host de la instancia.
    - Los datos se pierden al detener o cerrar la instancia.
    - Útiles para datos temporales o provisionales.

Se está ejecutando una instancia de Amazon EC2 con un almacén de instancias asociado.

La instancia se detiene o termina.

Se eliminan todos los datos del almacén de instancias asociado.



2. **Amazon Elastic Block Store (EBS)**:
    - Ofrece volúmenes de almacenamiento en bloque diseñados para datos que deben persistir.
    - Los datos permanecen disponibles incluso si la instancia EC2 se detiene o finaliza.
    - Se pueden personalizar el tamaño y tipo del volumen antes de asociarlo a una instancia EC2.
    - Admite copias de seguridad incrementales mediante **instantáneas**, para garantizar la seguridad y recuperación de datos.

Diferencia clave:

- **Almacenes de instancias** son temporales y vinculados a la instancia.
- **EBS** es persistente y adecuado para datos críticos.

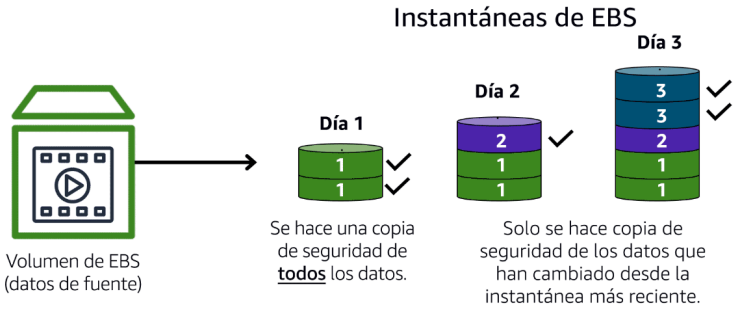
1.3 Instantáneas de Amazon EBS

- **Definición**:
  Una instantánea de Amazon EBS es una copia de seguridad incremental de un volumen de EBS.
- **Funcionamiento**:
    - La **primera instantánea** de un volumen copia todos los datos del mismo.
    - En **instantáneas posteriores**, solo se guardan los bloques de datos que han cambiado desde la instantánea anterior.
    - Este enfoque optimiza el uso de almacenamiento y reduce el tiempo necesario para realizar copias de seguridad.
- **Diferencia con copias completas**:
    - **Copias completas**: Replican todos los datos del volumen en cada respaldo, incluso aquellos que no han cambiado.
    - **Instantáneas incrementales**: Solo almacenan los cambios desde la última copia, haciendo el proceso más eficiente.

Ejemplo:

- Día 1: Se realiza la primera copia completa de dos volúmenes.
- Día 2: Se añade un nuevo volumen y solo este se respalda.
- Día 3: Se agregan dos volúmenes más y se respaldan únicamente estos.

Beneficio clave:
Ahorro de tiempo y espacio en comparación con las copias completas tradicionales.



Amazon Simple Storage Service (Amazon S3)

1.1 Introducción (resumen vídeo)

Amazon S3 es un servicio de almacenamiento en la nube que permite guardar y recuperar datos a cualquier escala. Organiza los datos como objetos dentro de buckets y ofrece múltiples clases de almacenamiento, desde acceso frecuente (S3 Standard) hasta archivo a largo plazo (S3 Glacier). Además, facilita la gestión automática del ciclo de vida de los datos, alta durabilidad y opciones como alojar sitios web estáticos. Es ideal para empresas que necesitan un almacenamiento seguro, flexible y escalable.

1.2 Almacenamiento de objetos

En el almacenamiento de objetos, cada elemento está compuesto por tres partes:

1. **Datos**: El contenido del archivo, como una imagen, video, documento, etc.
2. **Metadatos**: Información descriptiva sobre los datos, como su tipo, tamaño o cómo deben usarse.
3. **Clave**: Un identificador único que permite localizar el objeto dentro del sistema.

Esta estructura facilita una organización eficiente y accesible en servicios como Amazon S3.

1.3 Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento a nivel de objeto que organiza los datos en **buckets**. Permite cargar cualquier tipo de archivo (imágenes, videos, documentos, etc.) con un tamaño máximo de 5 TB por objeto y ofrece almacenamiento ilimitado.

Además, proporciona opciones como:

- **Control de permisos**: Para gestionar visibilidad y acceso.
- **Control de versiones**: Para rastrear cambios y preservar versiones anteriores de los objetos.

Ideal para copias de seguridad, archivos multimedia o almacenamiento de documentos archivados.

1.4 Clases de almacenamiento de Amazon S3

Amazon S3 ofrece diferentes clases de almacenamiento para adaptarse a diversas necesidades empresariales y económicas, considerando la frecuencia de acceso y la disponibilidad requerida para los datos.

1. **S3 Standard**
    - Para datos de acceso frecuente.
    - Alta disponibilidad y durabilidad en tres zonas de disponibilidad.
    - Ideal para sitios web, análisis y distribución de contenido.
2. **S3 Standard-Infrequent Access (S3 Standard-IA)**
    - Para datos de acceso poco frecuente pero alta disponibilidad.
    - Menor coste de almacenamiento, pero mayor coste de recuperación.
3. **S3 One Zone-Infrequent Access (S3 One Zone-IA)**
    - Almacenamiento en una sola zona de disponibilidad.
    - Más económico, pero con menor resiliencia ante fallos.
4. **S3 Intelligent-Tiering**
    - Optimiza costos automáticamente según patrones de acceso.
    - Cambia entre niveles de acceso frecuente e infrecuente.
5. **S3 Glacier Instant Retrieval**
    - Para datos archivados con acceso inmediato.
    - Recuperación en milisegundos.
6. **S3 Glacier Flexible Retrieval**
    - Archivos con acceso menos urgente.
    - Recuperación en minutos u horas, a menor costo.
7. **S3 Glacier Deep Archive**
    - Almacenamiento de bajo coste para datos a largo plazo.
    - Recuperación en 12 a 48 horas.
8. **S3 Outposts**

    - Almacenamiento local en entornos de AWS Outposts.
    - Para cargas de trabajo con requisitos de datos locales.

Cada clase está diseñada para optimizar costos y rendimiento según el uso y la criticidad de los datos.

1.5 Comparativa entre Amazon EBS y Amazon S3

En este combate entre servicios de almacenamiento, Amazon S3 y Amazon EBS se destacan por sus características únicas:

- **Amazon S3**:
    - Almacenamiento ilimitado y regional.
    - Ideal para objetos como imágenes, vídeos o documentos.
    - Diseñado para cargas de trabajo donde se accede y consume el objeto completo, con alta durabilidad y sin necesidad de instancias EC2.
    - Perfecto para aplicaciones web y almacenamiento de datos masivos con acceso concurrente.
- **Amazon EBS**:
    - Almacenamiento en bloques hasta 16 TiB.
    - Ideal para cargas de trabajo que requieren actualizaciones frecuentes y rápidas a nivel de bloque, como la edición de archivos grandes.
    - Dependiente de instancias EC2 para su funcionamiento.

Conclusión:
El ganador depende de tus necesidades. Usa **S3** para almacenamiento de objetos y cargas ocasionales, y **EBS** para tareas con cambios complejos y acceso directo a bloques.

Amazon Elastic File System (Amazon EFS)

1.1 Amazon EFS VS EBS

Resumen: Amazon EFS vs. EBS

- **Amazon EFS**:
    - Sistema de archivos compartido y administrado, ideal para varias instancias EC2 accediendo simultáneamente.
    - Escala automáticamente según las necesidades, sin aprovisionamiento manual.
    - Funciona a nivel regional y está optimizado para sistemas de archivos Linux.

- **Amazon EBS**:
    - Almacenamiento en bloques, adjunto a una única instancia EC2 dentro de una misma zona de disponibilidad (AZ).
    - No escala automáticamente, requiere aprovisionamiento manual.
    - Similar a un disco duro físico.

Diferencia clave:
EFS permite múltiples accesos concurrentes y escalado automático, mientras que EBS es más limitado en alcance y capacidad de escalado.

1.2 Almacenamiento de archivos

El almacenamiento de archivos permite que varios clientes (usuarios, aplicaciones, servidores, etc.) accedan a datos organizados en carpetas compartidas mediante rutas de archivos. Utiliza almacenamiento en bloques con un sistema de archivos local para gestionar los datos.

Características principales:

- Ideal para escenarios donde muchos servicios necesitan acceder simultáneamente a los mismos datos.
- Amazon Elastic File System (EFS) es un sistema de archivos escalable que crece o se reduce automáticamente según la demanda, alcanzando petabytes sin interrumpir aplicaciones.

1.3 Comparación entre Amazon EBS y Amazon EFS

Amazon EBS: Un volumen de Amazon EBS almacena datos en una **única** zona de disponibilidad.

Para asociar una instancia de Amazon EC2 a un volumen de EBS, ambos deben residir en la misma zona de disponibilidad.

Amazon EFS: Amazon EFS es un servicio regional. Almacena datos transversalmente en **varias** zonas de disponibilidad.
El almacenamiento duplicado permite acceder a los datos simultáneamente desde todas las zonas de disponibilidad de la región donde se encuentra un sistema de archivos. Además, los servidores en las instalaciones pueden acceder a Amazon EFS mediante AWS Direct Connect.

Amazon Relational Database Service (Amazon RDS)

1.1 Base de datos relacional

Las bases de datos relacionales almacenan datos de manera que se pueden relacionar entre sí. Utilizan tablas para organizar la información, y cada registro dentro de la tabla contiene datos relacionados. Por ejemplo, en un sistema de gestión de inventarios de una cafetería, cada registro de la base de datos podría almacenar información sobre un artículo, como nombre, tamaño y precio.

Estas bases de datos emplean **SQL** (Structured Query Language) para almacenar y consultar datos, lo que facilita la organización y recuperación eficiente de la información. Este enfoque permite realizar consultas complejas para obtener información específica. Por ejemplo, los propietarios de la cafetería pueden usar SQL para identificar a los clientes que más compran un café con leche mediano.

Ejemplo de datos en una base de datos relacional:

- **Tabla de productos**: Nombre del producto, tamaño, precio, categoría.
- **Tabla de clientes**: Nombre, dirección, correo electrónico, historial de compras.
- **Tabla de ventas**: Fecha de la venta, ID del cliente, productos comprados, monto total.

Este modelo permite gestionar grandes cantidades de datos de forma estructurada y escalable.

ID	Nombre de producto	Tamaño	Precio
1	Café molido tostado medio	340 g	5,30 USD
2	Café molido tostado fuerte	567 g	9,27 USD

1.2 Amazon Relational Database Service

Amazon RDS es un servicio administrado que facilita la ejecución de bases de datos relacionales en la nube de AWS. Automiza tareas comunes de gestión de bases de datos, como aprovisionamiento de hardware, configuración, aplicación de parches y copias de seguridad. Esto permite a los usuarios centrarse en la innovación de sus aplicaciones, sin tener que gestionar tareas administrativas complejas.

RDS es compatible con varios motores de bases de datos populares, como MySQL, PostgreSQL, Oracle y SQL Server. Además, se integra con otros servicios de AWS, como AWS Lambda, para realizar consultas de bases de datos desde aplicaciones sin servidor.

Seguridad: Amazon RDS ofrece opciones de cifrado tanto en reposo (para proteger los datos almacenados) como en tránsito (para proteger los datos durante su envío y recepción).

1.3Motores de base de datos de Amazon RDS

Amazon RDS está disponible en seis motores de base de datos, que optimizan la memoria, el rendimiento o la entrada/salida (E/S). Los motores de base de datos compatibles incluyen:

- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB
- Oracle Database
- Microsoft SQL Server

1.4 Amazon Aurora

Amazon Aurora es una base de datos relacional de alta gama compatible con MySQL y PostgreSQL. Ofrece un rendimiento hasta cinco veces superior al de MySQL estándar y tres veces superior al de PostgreSQL estándar.

Aurora está diseñada para reducir costos operativos al minimizar las operaciones de entrada/salida (E/S) innecesarias, garantizando a la vez fiabilidad y disponibilidad de los recursos de la base de datos.

Es ideal para cargas de trabajo que requieren alta disponibilidad, ya que replica seis copias de los datos en tres zonas de disponibilidad y realiza copias de seguridad continuas en Amazon S3.

Amazon DynamoDB

1.1 Introducción (resumen vídeo)

Amazon DynamoDB es una base de datos NoSQL sin servidor, lo que significa que no necesitas gestionar las instancias ni la infraestructura subyacente. Ofrece tablas donde se almacenan y consultan datos, organizados en elementos con atributos. DynamoDB maneja automáticamente el almacenamiento y el escalado, tanto horizontal como vertical, sin intervención del usuario.

Se caracteriza por su alta disponibilidad, ya que replica los datos de manera redundante en todas las zonas de disponibilidad, y su alto rendimiento, con tiempos de respuesta en milisegundos. Es ideal para aplicaciones que requieren escalabilidad y fiabilidad, incluso con millones de usuarios.

A diferencia de las bases de datos SQL, DynamoDB no usa el lenguaje SQL y permite esquemas más flexibles, donde los elementos pueden tener atributos diferentes. Esto la hace ideal para datos con variaciones. Sin embargo, no admite consultas SQL complejas y se enfoca en consultas sencillas basadas en claves específicas.

DynamoDB es ideal para situaciones que requieren gran escalabilidad, como en el Prime Day de Amazon 2019, cuando gestionó miles de millones de solicitudes sin necesidad de gestión manual de la base de datos.

1.2 Bases de datos no relacionales

En las bases de datos no relacionales, los datos se almacenan en tablas, pero a diferencia de las bases de datos relacionales, estas no siguen la estructura tradicional de filas y columnas. En su lugar, utilizan estructuras más flexibles, como los *pares clave-valor*, donde los datos se organizan en elementos (claves) que tienen atributos (valores).

Las bases de datos no relacionales permiten añadir o eliminar atributos de los elementos sin restricciones, y no todos los elementos en la tabla tienen que tener los mismos atributos. Esta flexibilidad las hace ideales para datos dinámicos y estructuras que cambian con el tiempo.

Un ejemplo típico de este tipo de base de datos es Amazon DynamoDB, que utiliza esta estructura para ofrecer escalabilidad, rendimiento y flexibilidad sin necesidad de gestionar la infraestructura subyacente.

Ejemplo de datos de una base de datos no relacional:

Clave	Valor
1	**Nombre**: Juan Pérez **Dirección**: Calle Cualquiera 123 **Bebida favorita**: Café con leche mediano
2	**Nombre**: María García **Dirección**: Calle Principal 100 **Fecha de nacimiento**: 5 de julio de 1994

1.3 Amazon DynamoDB

Amazon DynamoDB es un servicio de base de datos clave-valor. Ofrece un rendimiento de milisegundos de un solo dígito a cualquier escala.

1. DynamoDB es sin servidor, por lo que no tienes que aprovisionar, aplicar parches ni administrar servidores. Tampoco tienes que instalar, mantener ni operar software.

2. A medida que el tamaño de la base de datos se reduce o aumenta, DynamoDB escala automáticamente para ajustarse a los cambios de capacidad y, al mismo tiempo, mantener un rendimiento constante. Por eso es una buena opción para los casos prácticos que requieren un alto rendimiento durante el escalado.

1.4 Amazon RDS VS Amazon DynamoDB

En el enfrentamiento entre **Amazon RDS** (relacional) y **Amazon DynamoDB** (NoSQL), cada uno tiene sus fortalezas según las necesidades específicas.

- **Amazon RDS**: Ideal para **análisis complejos** y **bases de datos que requieren relaciones entre varias tablas**. En escenarios como sistemas de gestión de cadenas de suministro que

necesitan uniones relacionales complejas, RDS es el ganador, ya que está diseñado para estos fines.

- **Amazon DynamoDB**: Es una base de datos **NoSQL** que utiliza un esquema de **clave-valor**, lo que la hace adecuada para casos de uso más simples, como almacenar listas de contactos. A pesar de no ser tan compleja como las bases de datos relacionales, DynamoDB es más eficiente y económica para cargas de trabajo que no requieren funcionalidades complejas, destacándose por su **escalabilidad** y **rendimiento rápido**.

En resumen, la elección entre RDS y DynamoDB depende de las necesidades específicas de la carga de trabajo de tu empresa. Cada uno tiene un uso ideal según el tipo de datos y la complejidad de las consultas que necesites.

Amazon Redshift

1.1 Amazon Redshift

A veces, las necesidades empresariales no se centran en datos en tiempo real, sino en el análisis de datos históricos. Las bases de datos tradicionales, aunque efectivas para transacciones rápidas y consultas en tiempo real, no son ideales para gestionar grandes volúmenes de datos históricos o complejos. Esto es especialmente relevante en contextos como inteligencia empresarial (BI) o análisis de big data.

Para estos casos, **los almacenes de datos** (Data Warehouse) son la solución adecuada, ya que están diseñados para gestionar grandes volúmenes de datos históricos, como ventas pasadas o producción, en lugar de datos operativos que cambian continuamente.

Un ejemplo de servicio ideal para esto es **Amazon Redshift**, un **almacén de datos como servicio**. Redshift es escalable, con la capacidad de manejar petabytes de datos y permite realizar consultas SQL sobre grandes conjuntos de datos almacenados en lagos de datos. Ofrece un rendimiento diez veces superior al de bases de datos tradicionales para cargas de trabajo de BI, lo que permite obtener respuestas rápidas y reducir el tiempo de espera.

En resumen, si necesitas soluciones de BI para analizar grandes volúmenes de datos históricos, **Amazon Redshift** es una opción excelente.

AWS Database Migration Service

1.1 Introducción (resumen vídeo)

Amazon Database Migration Service (DMS) facilita la migración de bases de datos a AWS, permitiendo que la base de datos fuente siga operativa durante el proceso. Existen dos tipos de migración:

1. **Homogénea**: Para bases de datos del mismo tipo (p. ej., MySQL a Amazon RDS for MySQL).
2. **Heterogénea**: Para bases de datos de tipos diferentes, usando AWS Schema Conversion Tool para adaptar los esquemas.

DMS también se puede usar para migrar a entornos de desarrollo, consolidar bases de datos o replicar datos continuamente.

1.2 AWS Database Migration Service (AWS DMS)

AWS Database Migration Service (AWS DMS) facilita la migración de bases de datos relacionales, no relacionales y otros almacenes de datos. Permite mover datos entre bases de datos de origen y destino, que pueden ser del mismo tipo o diferentes. Durante la migración, la base de datos fuente permanece operativa, reduciendo el tiempo de inactividad de las aplicaciones dependientes.

Por ejemplo, si tienes una base de datos MySQL en tus instalaciones, puedes migrar los datos a una base de datos como Amazon Aurora utilizando AWS DMS.

1.3 Otros casos prácticos de AWS DMS

Desarrollo y migración de base de datos de prueba: Permitir a los desarrolladores probar aplicaciones con datos de producción sin que afecte a los usuarios de producción.

Consolidación de bases de datos:Combinación de varias bases de datos en una única base de datos.

Replicación continua: Envío de copias continuas de tus datos a otras fuentes de destino en lugar de realizar una migración única

Servicios de bases de datos adicionales

1.1 Introducción (resumen vídeo)

El vídeo destaca la importancia de elegir la base de datos y plataforma de almacenamiento adecuadas para las necesidades específicas de un negocio, en lugar de ajustar los datos a la base de datos. AWS ofrece una variedad de bases de datos especializadas, como DynamoDB para pares clave-valor, DocumentDB para contenido y catálogos, Neptune para redes sociales y análisis de grafos, y QLDB para registros inmutables. También se menciona Amazon Managed Blockchain y soluciones como Amazon ElastiCache y DAX para mejorar la velocidad de lectura. En resumen, AWS se enfoca en proporcionar las herramientas más adecuadas para cada necesidad.

1.2 Servicios de datos adicionales

1. **Amazon DocumentDB:** Base de datos de documentos compatible con cargas de trabajo de MongoDB, ideal para gestionar documentos.
2. **Amazon Neptune:** Base de datos de grafos para aplicaciones que trabajen con datos altamente conectados, como motores de recomendaciones y detección de fraude.
3. **Amazon Quantum Ledger Database (QLDB)**: Base de datos de libro mayor para realizar un seguimiento completo de todos los cambios en los datos de una aplicación, garantizando inmutabilidad.
4. **Amazon Managed Blockchain:** Servicio para crear y gestionar redes de blockchain con marcos de trabajo de código abierto, permitiendo transacciones sin una autoridad central.
5. **Amazon ElastiCache**: Servicio que mejora los tiempos de lectura al añadir capas de almacenamiento en caché a las bases de datos, compatible con Redis y Memcached.
6. **Amazon DynamoDB Accelerator (DAX):** Caché en memoria para DynamoDB que reduce los tiempos de respuesta de milisegundos a microsegundos.

RESUMEN MÓDULO 5

Has aprendido sobre diversos mecanismos de almacenamiento en AWS:

1. **EBS**: Volúmenes de almacenamiento para instancias EC2.
2. **S3**: Almacenamiento de objetos mediante clics o APIs.
3. **Bases de datos relacionales**: Soluciones para bases de datos tradicionales.
4. **DynamoDB**: Base de datos no relacional para pares clave-valor.
5. **EFS**: Sistema de archivos para almacenamiento compartido.
6. **Redshift**: Almacenamiento y análisis de datos masivos.
7. **DMS**: Herramienta para migrar bases de datos.
8. **DocumentDB, Neptune, QLDB, Blockchain**: Servicios especializados en bases de datos de documentos, grafos, registros inmutables y blockchain.
9. **ElastiCache y DAX**: Soluciones de caché para mejorar el rendimiento.

.