COMPENSACIONES DE DISEÑO

- Evalúe las compensaciones para que pueda seleccionar un enfoque óptimo.
- Algunos ejemplos de compensación son:
  - Cambiar la consistencia, la durabilidad y el espacio por el tiempo y la latencia para ofrecer un rendimiento más alto.
  - Priorizar la velocidad de comercialización de las funciones nuevas por sobre el costo.
- Basar las decisiones de diseño en datos empíricos.

A medida que diseña una solución, piense detenidamente en las compensaciones para que pueda seleccionar un enfoque óptimo. Por ejemplo, podría cambiar la consistencia, la durabilidad y el espacio por el tiempo y la latencia para ofrecer un rendimiento más alto. O bien, podría priorizar la velocidad de comercialización por sobre el costo.

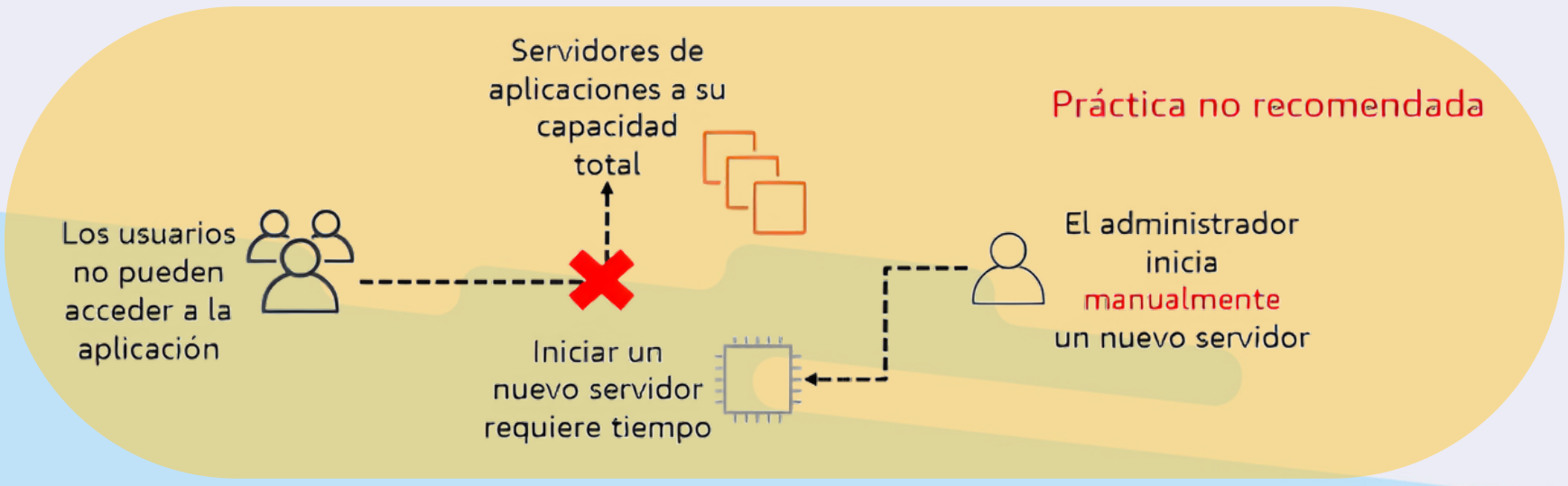
Las compensaciones pueden aumentar el costo y la complejidad de su arquitectura, por lo que las decisiones de diseño deben basarse en datos empíricos. Por ejemplo, podría tener que realizar pruebas de carga para asegurarse de obtener un beneficio medible del rendimiento. O bien, podría tener que realizar análisis comparativos para lograr, con el tiempo, la carga de trabajo con los mejores costos. Cuando evalúe las mejoras relacionadas con el rendimiento, también es deseable que considere de qué manera sus elecciones de diseño de arquitectura afectarán a sus clientes y las eficiencias de las cargas de trabajo.

En esta sección, conocerá las prácticas recomendadas para el diseño de soluciones en AWS. También conocerá las prácticas no recomendadas (o los malos diseños de soluciones) que debe evitar.

HABILITE LA EXCALABILIDAD (1 DE 2)

1

Asegúrese de que su arquitectura pueda manejar los cambios en la demanda



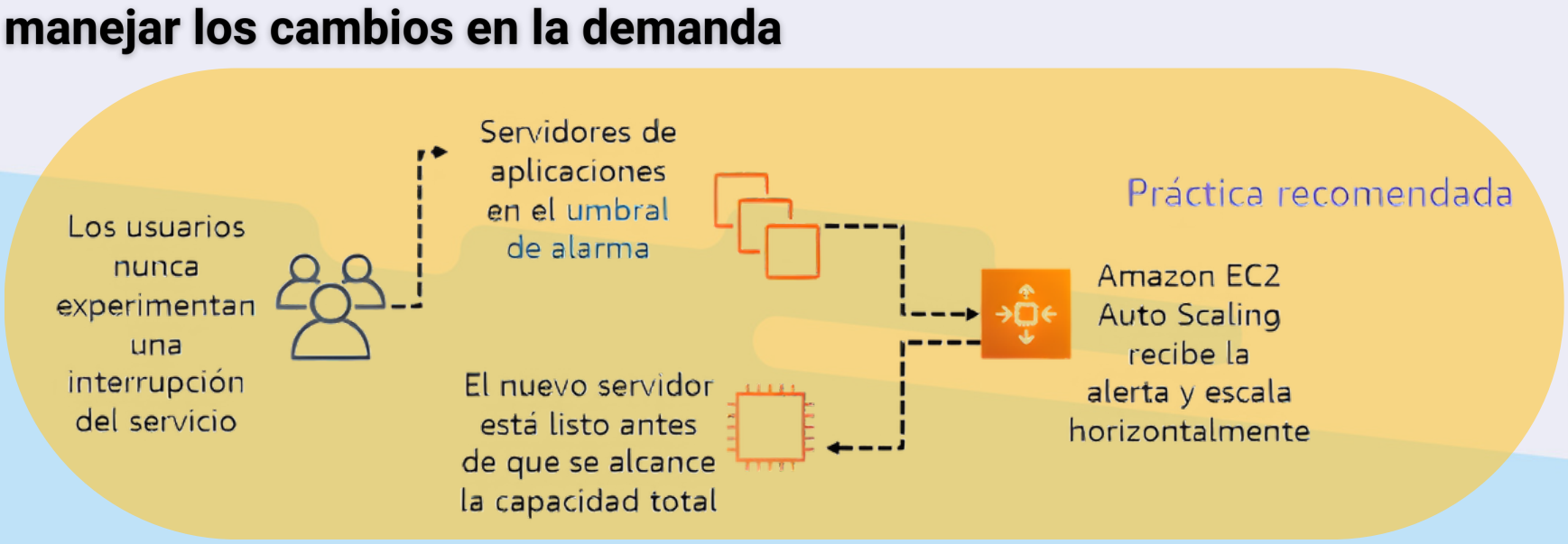
Cuando ejecuta sus cargas de trabajo en la nube de AWS, puede escalar su infraestructura con rapidez y de manera proactiva. Asegúrese de implementar la escalabilidad en cada capa de su infraestructura.

Para comprender la importancia de la escalabilidad, considere estas prácticas no recomendadas, donde el escalado se hace de manera reactiva y manual. En esta situación, cuando los servidores de aplicaciones alcanzan su capacidad total, se impide a los usuarios acceder a la aplicación. Luego, los administradores lanzarán manualmente una o más instancias nuevas para administrar la carga.

Lamentablemente, se requieren algunos minutos para que la instancia esté disponible para uso después de lanzarla. Eso aumenta el tiempo en que los usuarios no pueden acceder a la aplicación.

HABILITE LA EXCALABILIDAD (2 DE 2)

Asegúrese de que su arquitectura pueda manejar los cambios en la demanda



Al habilitar la escalabilidad, puede mejorar su diseño para anticipar la necesidad de más capacidad y entregarla antes de que sea demasiado tarde.

Por ejemplo, puede usar una solución de supervisión como **Amazon CloudWatch** para detectar si la carga total de toda su flota de servidores alcanzó un umbral especificado. Puede definir este umbral para que sea “Se mantuvo sobre el 60% de uso de la CPU durante más de 5 minutos”, o cualquier otra idea relacionada con el uso de los recursos. Con CloudWatch, también puede diseñar métricas personalizadas basadas en aplicaciones específicas que pueden activar el escalado de recursos que se requiere.

Amazon CloudWatch

Amazon EC2

Cuando se activa una alarma, **Amazon EC2 Auto Scaling** lanza de inmediato una nueva instancia. De este modo, esa instancia está lista antes de que se alcance el máximo de la capacidad, lo que ofrece una experiencia sin interrupciones para los usuarios.

Idealmente, también debe diseñar su sistema para que haga una reducción horizontal cuando la demanda disminuya y usted no ejecute (y pague) instancias que ya no necesita.

AUTOMATICE SU ENTORNO

2

AWS ofrece herramientas integradas de supervisión y automatización en casi todas las capas de su infraestructura. Aproveche estas herramientas para asegurarse de que su infraestructura pueda responder rápidamente a los cambios.
Puede usar herramientas como CloudWatch y Amazon EC2 Auto Scaling para detectar los recursos en mal estado y automatizar el lanzamiento de recursos de reemplazo. También puede recibir notificaciones cuando cambien las asignaciones de recursos.

TRATE LOS RECURSOS COMO DESECHABLES

3

Aproveche la naturaleza de aprovisionamiento dinámico del cómputo en la nube.

PRÁCTICA RECOMENDADA

- Automatice la implementación de recursos nuevos con configuraciones idénticas
- Termine los recursos que no se utilizan
- Cambie a nuevas direcciones IP de manera automática
- Pruebe las actualizaciones en los recursos nuevos, luego, reemplace los recursos viejos por los actualizados

PRÁCTICA NO RECOMENDADA

- Con el tiempo, servidores distintos terminan teniendo configuraciones diferentes
- Los recursos se ejecutan cuando no se necesitan
- Las direcciones IP codificadas de manera rígida impiden la flexibilidad
- Puede ser difícil o inconveniente probar actualizaciones nuevas en hardware que está en uso

La práctica recomendada de tratar los recursos como desechables se relaciona con la idea de considerar su infraestructura como software en lugar de como hardware.
Con el hardware, es fácil comprar más componentes específicos de los que necesita como preparación para los picos de uso. Eso es costoso y carece de flexibilidad. Es más difícil de actualizar debido a los costos irrecuperables.

En cambio, cuando trata los recursos como desechables, migrar entre instancias u otros recursos discretos es bastante sencillo. Puede responder con rapidez a los cambios en las necesidades de capacidad, actualizar aplicaciones y administrar el software subyacente.

UTILICE COMPONENTES DE ACOPLAMIENTO DÉBIL

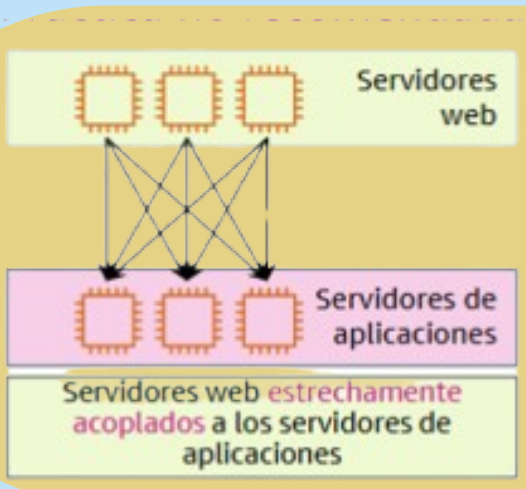
4

Diseñe arquitecturas con componentes independientes.

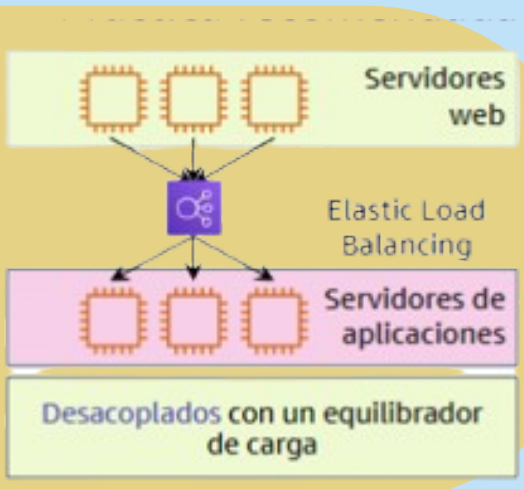
Las infraestructuras tradicionales se basan en cadenas de servidores estrechamente integrados, cada uno con un propósito específico. El problema es que cuando uno de esos componentes o capas deja de funcionar, la interrupción del sistema puede ser crítica. También impide el escalado. Si agrega o quita servidores en una capa, también debe conectar todos los servidores de cada capa de conexión.

El ejemplo de la izquierda muestra un conjunto de servidores web y de aplicaciones que están estrechamente acoplados. Si un servidor de aplicaciones deja de funcionar, se producirá un error porque los servidores web intentarán conectarse a él sin éxito.
Con el acoplamiento débil, utiliza soluciones administradas como intermediarios entre las capas de su sistema. Con este diseño, el intermediario automáticamente maneja ambos errores y el escalado de los componentes o las capas.

Práctica no recomendada



Práctica recomendada



El ejemplo de la derecha muestra un equilibrador de carga (en este caso, un equilibrador de carga de Elastic Load Balancing) que enruta las solicitudes entre los servidores web y los servidores de aplicaciones. Si un servidor de aplicaciones deja de funcionar, el equilibrador de carga automáticamente comenzará a dirigir todo el tráfico a los dos servidores en buen estado.

Dos soluciones primarias para desacoplar los componentes son los **equilibradores de carga** y las **colas de mensajes**.

DISEÑE SERVICIOS, NO SERVIDORES

5

Use la variedad de servicios de AWS. No limite su infraestructura a servidores.

PRÁCTICA RECOMENDADA

- Cuando corresponda, considere usar contenedores o una solución sin servidor
- Las colas de mensajes manejan la comunicación entre aplicaciones
- Los activos web estáticos se almacenan de manera externa, como en Amazon Simple Storage Service (Amazon S3)
- La autenticación de usuarios y el almacenamiento del estado de usuarios son manejados por los servicios administrados de AWS

PRÁCTICA NO RECOMENDADA

- Las aplicaciones simples se ejecutan en servidores persistentes
- Las aplicaciones se comunican directamente entre sí
- Los activos web estáticos se almacenan localmente en las instancias
- Los servidores backend manejan la autenticación de usuarios y el almacenamiento del estado de usuarios

La siguiente práctica recomendada es diseñar servicios, no servidores. Aunque **Amazon Elastic Compute Cloud (Amazon EC2)** ofrece una enorme flexibilidad para diseñar y configurar su solución, no siempre debe ser la primera (o la única) solución que utiliza para cada necesidad. En algunos casos, los contenedores o una solución sin servidor podrían ser más adecuados. Por lo tanto, es importante considerar cuáles son sus necesidades y cuál solución es adecuada.

Con las soluciones sin servidor y los servicios administrados de AWS, no se necesita aprovisionar, configurar y administrar toda una instancia de Amazon EC2. Las soluciones administradas que tienen un perfil más bajo y un mejor rendimiento pueden reemplazar las soluciones basadas en servidor por un menor costo. Algunos ejemplos son **AWS Lambda, Amazon Simple Queue Service (Amazon SQS), Amazon DynamoDB, Elastic Load Balancing, Amazon Simple Email Service (Amazon SES) y Amazon Cognito**.

ELIJA LA SOLUCIÓN DE BASE DE DATOS ADECUADA

Adapte la tecnología a la carga de trabajo, no al revés

Aspectos para tener en cuenta

- Las necesidades de lectura y escritura
- Los requisitos de almacenamiento totales
- El tamaño habitual de los objetos y cómo se accede a ellos
- Los requisitos de durabilidad
- Los requisitos de latencia
- El número máximo de usuarios conectados al mismo tiempo
- La naturaleza de las consultas
- La intensidad necesaria de los controles de integridad

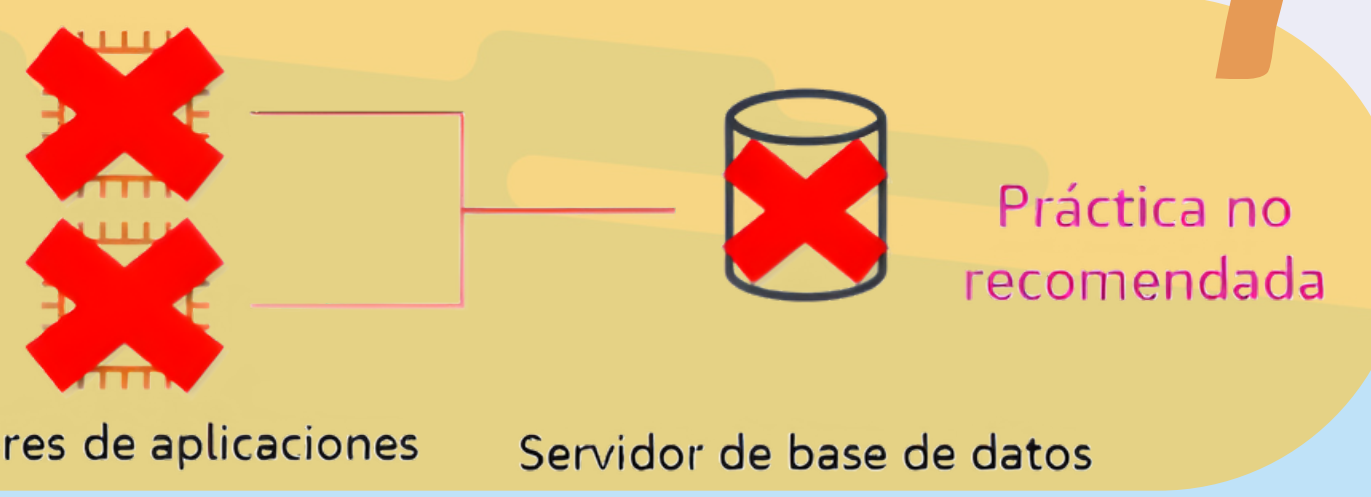
Es importante que elija la solución de base de datos adecuada. En los centros de datos tradicionales y los entornos en las instalaciones, los límites sobre el hardware y las licencias disponibles pueden resultar una restricción al momento de elegir una solución de almacén de datos. AWS recomienda elegir un almacén de datos basado en las necesidades para su entorno de aplicaciones.

EVITE LOS PUNTOS ÚNICOS DE ERROR (1 DE 2)

Suponga que todo falla. Luego,
diseñe en sentido inverso

Cuando sea posible, use la redundancia para impedir que los puntos únicos hagan fallar un sistema completo.

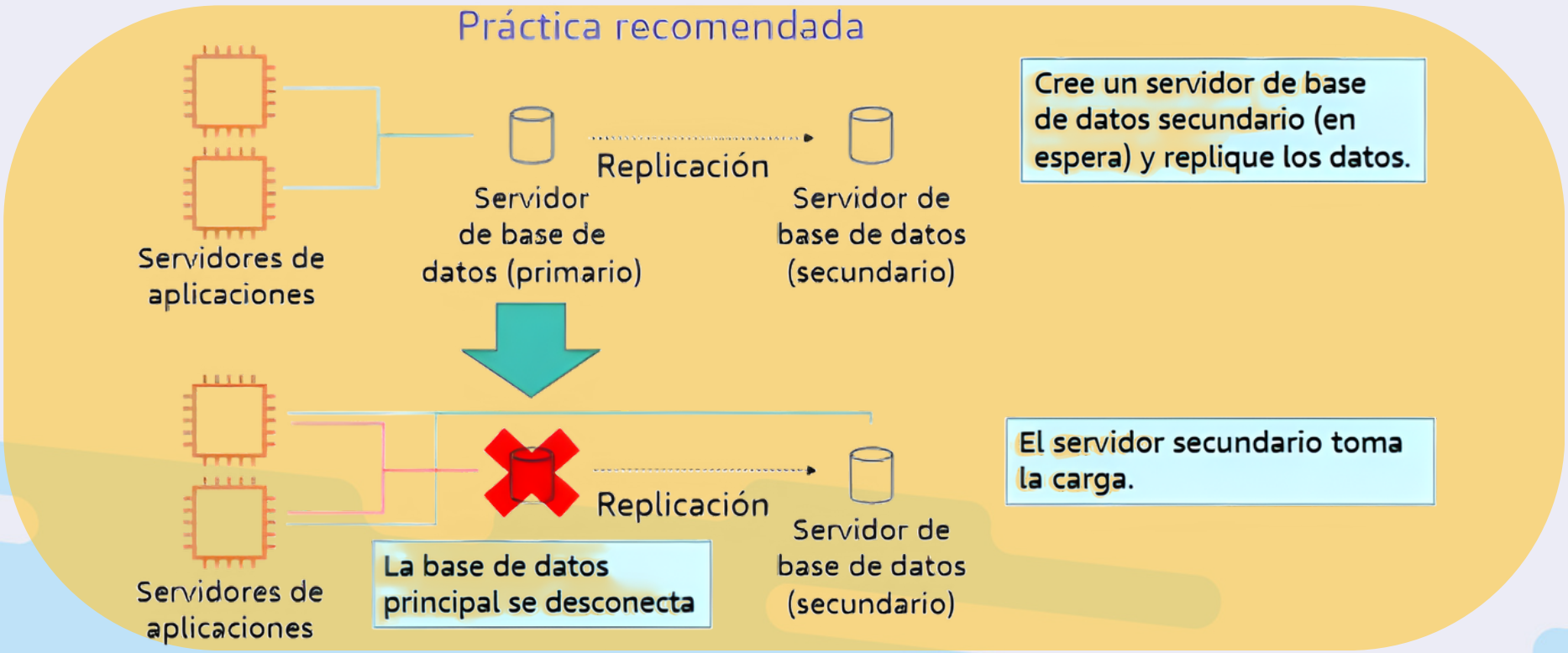
7



Cuando sea posible, elimine los puntos únicos de error de su arquitectura. Esto no quiere decir que no debe duplicar siempre cada componente. Dependiendo de sus acuerdos de nivel de servicio **(SLA, Service-Level Agreement)** de tiempo de inactividad, puede usar soluciones automatizadas que solo inician componentes cuando es necesario. También puede usar un servicio administrado, donde AWS reemplaza automáticamente el hardware defectuoso subyacente por usted.

Este sistema simple muestra dos servidores de aplicaciones conectados a un único servidor de base de datos. El servidor de base de datos representa un punto único de error y se debe evitar. Cuando deja de funcionar, los servidores de aplicaciones también dejan de hacerlo. Los servidores de aplicaciones deben seguir funcionando incluso en caso de falla, eliminación o sustitución del hardware físico subyacente.

EVITE LOS PUNTOS ÚNICOS DE ERROR (2 DE 2)



Una forma común de evitar los puntos únicos de error es crear un servidor de base de datos secundarios (en espera) y replicar los datos. De este modo, si el servidor de base de datos principal se desconecta, el servidor secundario puede asumir la carga.
En este ejemplo, cuando la base de datos principal se desconecta, los servidores de aplicaciones envían automáticamente sus solicitudes a la base de datos secundaria. Este ejemplo también ilustra la práctica recomendada 3: trate los recursos como desechables y diseñe sus aplicaciones para que admitan cambios en el hardware.

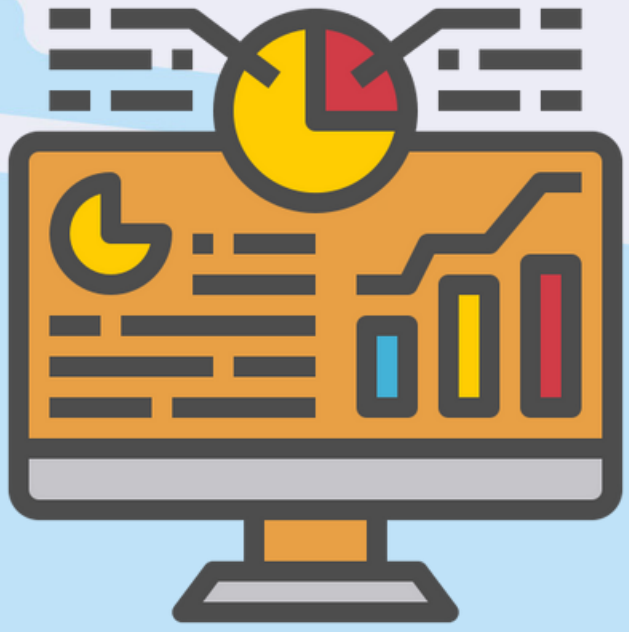
OPTIMICE EL COSTO

Aproveche la flexibilidad de AWS para aumentar su eficiencia de costos.

8

Aspectos para tener en cuenta

- ¿Mis recursos son del tamaño y tipo adecuados para el trabajo?
- ¿Qué métricas debo supervisar?
- ¿Cómo me aseguro de desactivar los recursos que no están en uso?
- ¿Con qué frecuencia necesito usar este recurso?
- ¿Puedo reemplazar cualquiera de mis servidores con servidores administrados?



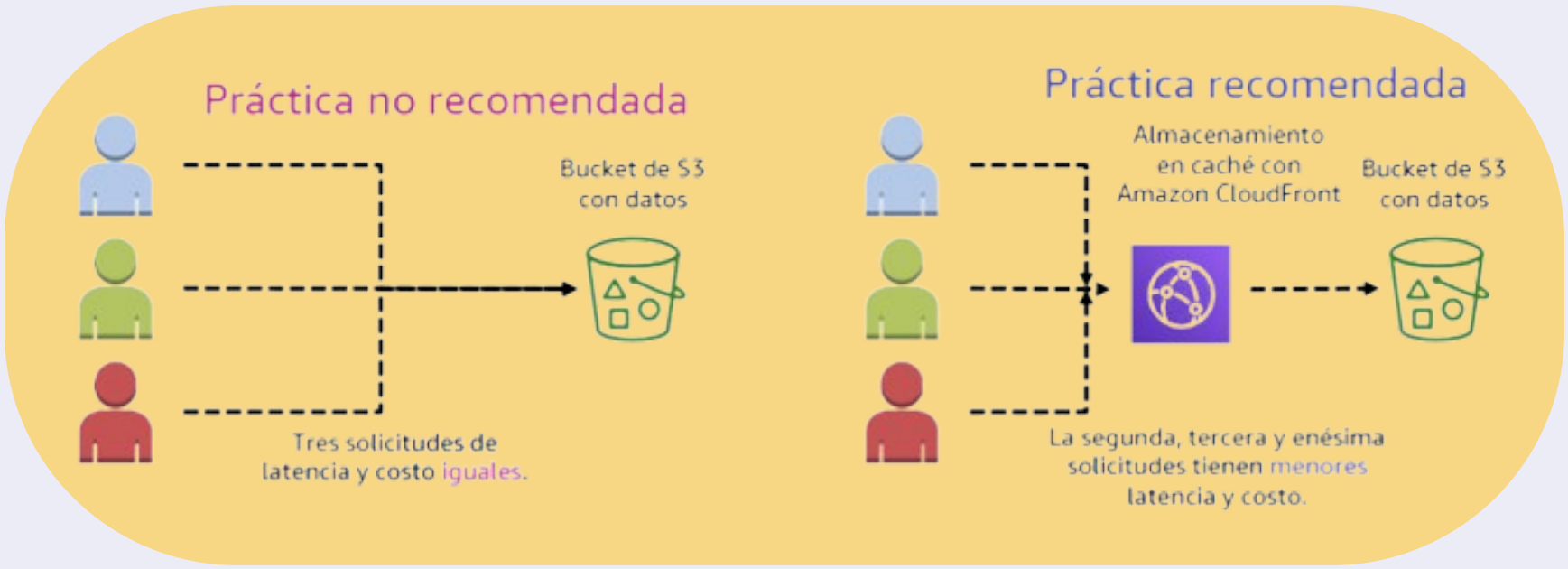
El cómputo en la nube le permite cambiar los gastos de capital por los gastos variables.
Los gastos de capital **(CapEx)** son fondos que utiliza una empresa para adquirir, actualizar y mantener activos físicos como bienes, edificios industriales o equipos. Según este modelo, usted paga por los servidores en el centro de datos, estén o no activos. Por el contrario, los servicios de AWS usan un modelo de costos de gastos variables, lo que significa que solo paga por los servicios individuales que necesita, por el tiempo que los utilice. En cada servicio, puede optimizar el costo. Muchos servicios ofrecen diferentes niveles de precios, modelos o configuraciones.

Tenga en cuenta que puede ser muy costoso replicar una configuración de centro de datos en las instalaciones de servidores que se ejecutan 24x7 en la nube. Por lo tanto, la mejor manera de construir su infraestructura, desde la perspectiva del costo, es aprovisionar únicamente los recursos que necesita y detener los servicios cuando no se utilicen.

USE EL ALMACENAMIENTO EN CACHÉ

El almacenamiento en caché minimiza las operaciones de recuperación de datos redundantes, lo que mejora el rendimiento y el costo.

El almacenamiento en caché es una técnica para hacer futuras solicitudes con más rapidez y reducir el rendimiento de la red almacenando temporalmente los datos en una ubicación intermedia entre el solicitante y el almacenamiento permanente. En el ejemplo de prácticas no recomendadas, no se utilizó ningún servicio de almacenamiento en caché. Cuando alguien solicita un archivo de uno de los buckets de **Amazon Simple Storage Service (Amazon S3)**, cada solicitud tarda la misma cantidad de tiempo en completarse y cada solicitud tiene el mismo costo.



En el ejemplo de prácticas recomendadas, la infraestructura usa **Amazon CloudFront** antes de Amazon S3 para ofrecer almacenamiento en caché. En esta situación, la solicitud inicial busca el archivo en Amazon CloudFront. Si no lo encuentra, CloudFront solicita el archivo a Amazon S3. Después, CloudFront almacena una copia del archivo en una ubicación perimetral cerca del usuario y envía una copia al usuario que hizo la solicitud. Las solicitudes posteriores del archivo se recuperan desde la (ahora más cercana) ubicación perimetral en CloudFront en lugar de Amazon S3. Esto reduce la latencia y el costo porque, después de la primera solicitud, ya no debe pagar por la transferencia del archivo desde Amazon S3.

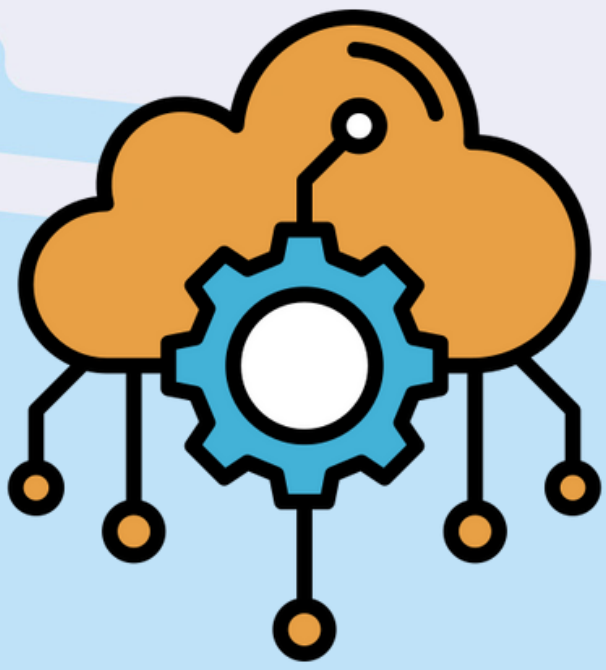
PROTEJA LA INFRAESTRUCTURA COMPLETA

Incorpore seguridad en cada capa de su infraestructura.

10

Aspectos para tener en cuenta

- Aislar partes de su infraestructura
- Cifrar los datos en tránsito y en reposo
- Aplicar la granularidad del control de acceso, mediante el principio de mínimo privilegio
- Utilizar la autenticación multifactor (MFA)
- Utilizar servicios administrados
- Registrar el acceso a recursos
- Automatizar las implementaciones para mantener la coherencia de la seguridad



La seguridad no solo consiste en atravesar el límite externo de la infraestructura. También implica garantizar que los entornos individuales y sus componentes estén protegidos entre sí. Por ejemplo, en Amazon EC2, puede crear grupos de seguridad que le permitan determinar cuáles puertos en sus instancias pueden enviar y recibir tráfico. Los grupos de seguridad también pueden determinar el origen y destino de ese tráfico. Puede usar grupos de seguridad para reducir la probabilidad de que una amenaza de seguridad en una instancia se propague a todas las demás instancias en su entorno. Debe tomar precauciones similares con otros servicios. Las maneras específicas de implementar esta práctica recomendada se analizarán durante el BootCamp.

LOS APRENDIZAJES CLAVE DE ESTA SECCIÓN DEL MÓDULO SON:

Cuando diseñe soluciones, evalúe las compensaciones y fundamente sus decisiones en datos empíricos:

Sigas estas prácticas recomendadas cuando cree soluciones en AWS:

- Habilite la escalabilidad
- Automatice su entorno
- Trate los recursos como desechables
- Utilice componentes de acoplamiento débil
- Diseñe servicios, no servidores
- Elija la solución de base de datos adecuada
- Evite los puntos únicos de error
- Optimice el costo
- Use el almacenamiento en caché
- Proteja la infraestructura completa