



training and  
certification

**AWS Academy Cloud Architecting (ES)  
Module 05 Student Guide  
Versión 2.0.2**

**200-ACACAD-20-ES-SG**

© 2021 Amazon Web Services, Inc. o sus empresas afiliadas.  
Todos los derechos reservados.

Este contenido no puede reproducirse ni redistribuirse, total ni parcialmente,  
sin el permiso previo por escrito de Amazon Web Services, Inc. Queda prohibida  
la copia, el préstamo o la venta de carácter comercial.

Para correcciones o comentarios relacionados con el curso,  
contacte con nosotros en:  
<https://support.aws.amazon.com/#/contacts/aws-training>.

Todas las marcas comerciales pertenecen a sus propietarios.

# Contenido

Módulo 5: Creación de la capa de base de datos

4

AWS Academy Cloud Architecting

# Módulo 5: Incorporación de una capa de base de datos



Bienvenido al Módulo 5: Incorporación de una capa de base de datos.

## Información general sobre el módulo



### Secciones

1. Necesidad de arquitectura
2. Consideraciones de la capa de base de datos
3. Amazon RDS
4. Amazon DynamoDB
5. Controles de seguridad de la base de datos
6. Migración de datos a las bases de datos de AWS

### Demostración

- Réplicas de lectura y copias de seguridad automatizadas de Amazon RDS

### Laboratorios

- Laboratorio guiado: Creación de una base de datos de Amazon RDS
- Laboratorio de desafíos: Migración de una base de datos a Amazon RDS



### Evaluación de conocimientos

Este módulo incluye las siguientes secciones:

1. Necesidad de arquitectura
2. Consideraciones de la capa de base de datos
3. Amazon RDS
4. Amazon DynamoDB
5. Controles de seguridad de la base de datos
6. Migración de datos a las bases de datos de AWS

Este módulo también incluye lo siguiente:

- Una demostración de la configuración automatizada de copias de seguridad y réplicas de lectura de Amazon Relational Database Service (Amazon RDS).
- Un laboratorio guiado donde puede crear una base de datos de Amazon RDS y conectarse a ella mediante una aplicación web sencilla.
- Un laboratorio de desafíos que lo desafía a migrar datos desde una base de datos que se ejecuta en una instancia de Amazon Elastic Compute Cloud (Amazon EC2) a una base de datos de Amazon RDS.

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

## Objetivos del módulo



Una vez finalizado este módulo, debería ser capaz de lo siguiente:

- Comparar tipos de bases de datos
- Diferenciar entre servicios administrados y no administrados
- Explicar cuándo usar Amazon Relational Database Service (Amazon RDS)
- Explicar cuándo usar Amazon DynamoDB
- Describir los controles de seguridad de la base de datos
- Describir cómo migrar datos a bases de datos de Amazon Web Services (AWS)
- Implementar un servidor de base de datos

Una vez finalizado este módulo, debería ser capaz de lo siguiente:

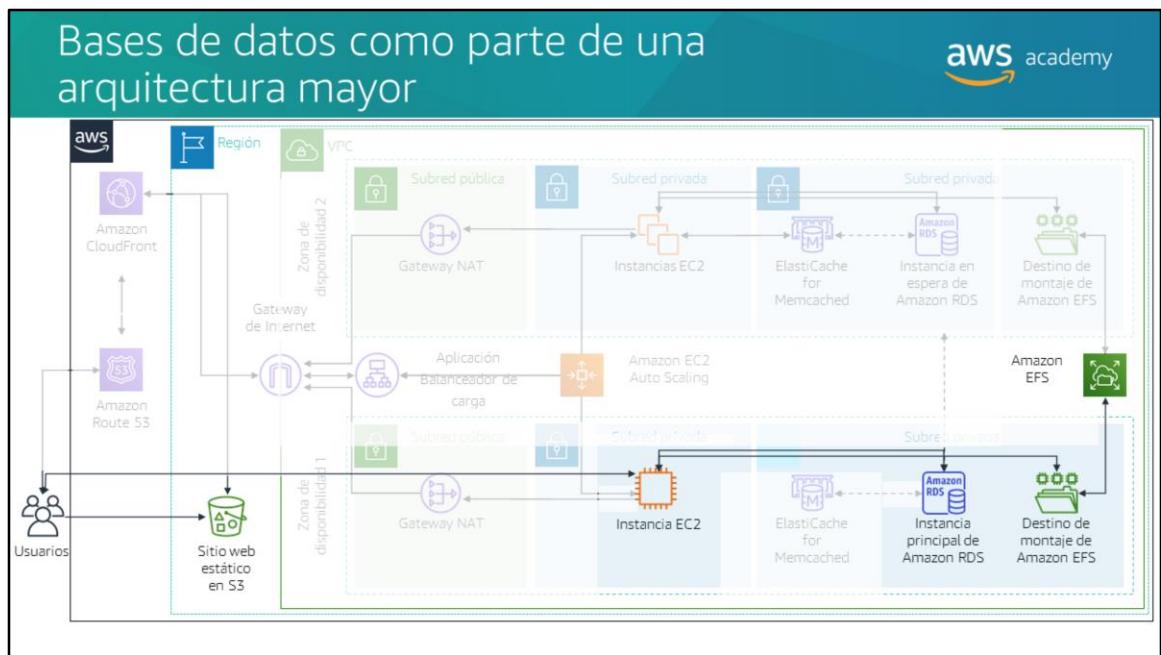
- Comparar tipos de bases de datos
- Diferenciar entre servicios administrados y no administrados
- Explicar cuándo usar Amazon Relational Database Service (Amazon RDS)
- Explicar cuándo usar Amazon DynamoDB
- Describir los controles de seguridad de la base de datos
- Describir cómo migrar datos a bases de datos de Amazon Web Services (AWS)
- Implementar un servidor de base de datos

**Módulo 5: Incorporación de una capa de base de datos**

**Sección 1: Necesidad de arquitectura**



Presentación de la Sección 1: Necesidad de arquitectura



A medida que cada módulo presente nuevas características, se darán a conocer esas partes de este diagrama más grande.

En este módulo, aprenderá cómo crear una arquitectura que utilice los servicios de base de datos de AWS. Se analizan diferentes opciones de servicio de bases de datos relacionales y no relacionales (incluido Amazon RDS). A medida que aprenda sobre los beneficios esenciales de las diversas ofertas, podrá tomar una decisión más informada sobre el servicio de base de datos que mejor se adapte a sus necesidades de arquitectura particulares.

## Requisito empresarial de la cafetería



La cafetería necesita una solución de base de datos que sea más fácil de mantener y que ofrezca características esenciales como durabilidad, escalabilidad y alto rendimiento.



Desde que la cafetería agregó la posibilidad de realizar pedidos en línea a su sitio web, el personal se dio cuenta de que la actividad aumentó. Además, descubrieron que el historial de pedidos almacenado en la base de datos, que instalaron en la misma instancia EC2 en la que se ejecuta el servidor web, proporciona información valiosa para la empresa. Martha lo usa para fines contables, y Frank lo mira de vez en cuando para hacerse una idea de la cantidad de cada tipo de postre que debe hornear.

Sin embargo, Sofía tiene algunas inquietudes. La base de datos debe actualizarse y se le deben aplicar parches, y ella no siempre tiene tiempo para realizar estas tareas de manera regular. Además, administrar la base de datos es una habilidad especializada, y capacitar a otros para que lo hagan no es algo a lo que ella quiere dedicar tiempo. Mientras tanto, también le preocupa que la cafetería no esté haciendo copias de seguridad de los datos con la frecuencia que se debería.

El personal de la cafetería quiere reducir los costos de mano de obra relacionados con la inversión en aprendizaje técnico que se necesita para administrar la base de datos ellos mismos. Por lo tanto, decidieron que necesitan usar una solución de base de datos administrada. Idealmente, encontrarán una que ofrezca características esenciales, como durabilidad, escalabilidad y alto rendimiento.

En este módulo, aprenderá detalles sobre los diferentes servicios de base de datos que ofrece AWS y las capacidades que proporcionan estos diferentes servicios. Ya que comprende las opciones disponibles, debería poder elegir una solución de base de datos que pueda satisfacer con éxito estos nuevos requisitos empresariales.

**Módulo 5: Incorporación de una capa de base de datos**

**Sección 2: Consideraciones de la capa  
de base de datos**



Presentación de la Sección 2: Consideraciones de la capa de base de datos.

## Consideraciones de la base de datos: escalabilidad



### Escalabilidad

¿Cuánto rendimiento se necesita?

¿La solución elegida será capaz de escalar en forma ascendente más adelante si resulta necesario?



Requisitos de almacenamiento totales



Tamaño y tipo de objeto



Durabilidad



Como arquitecto, a menudo tendrá que elegir entre diferentes tipos de bases de datos cuando considere qué tipo manejará mejor una carga de trabajo en particular. Antes de elegir una base de datos, hay algunas consideraciones clave que deberían aportar información al proceso de toma de decisiones.

En primer lugar, considere la importancia de la *escalabilidad*. Con las bases de datos tradicionales en las instalaciones, el escalado de la capacidad puede ser una tarea difícil incluso para los administradores de base de datos experimentados. Puede tomar horas, días o semanas. El impacto en el rendimiento de la base de datos mientras se escala puede ser impredecible y puede requerir tiempo de inactividad. Sin embargo, no se puede sobreestimar la importancia de una base de datos escalada correctamente. Si el aprovisionamiento de su base de datos es insuficiente, es posible que las aplicaciones dejen de funcionar. Sin embargo, si el aprovisionamiento de su base de datos es excesivo, aumentará los costos iniciales mediante la adquisición de recursos que no necesita, lo que infringe el principio de optimización de costos de AWS Well-Architected Framework.

Idealmente, debería elegir una solución de base de datos que tenga los recursos para manejar el *rendimiento* necesario en el lanzamiento y que también pueda *escalarse* en forma ascendente fácilmente más adelante si debe aumentar su rendimiento.

Otra funcionalidad con la que es agradable contar es la capacidad de escalar en forma descendente la capacidad de la base de datos aprovisionada si los requisitos de rendimiento o la carga de la base de datos disminuyen posteriormente. Esto le permite obtener ahorros de costos inmediatamente reduciendo la capacidad aprovisionada. Una solución de escalado automático podría reducir los costos de sistema y de mano de obra.

## Consideraciones de la base de datos: requisitos de almacenamiento



Escalabilidad

¿Qué tan grande debe ser la base de datos?

**Requisitos de almacenamiento totales**

¿Necesitará almacenar GB,  
TB o petabytes de datos?

Tamaño y tipo de objeto



Durabilidad

En segundo lugar, cuando debe elegir una base de datos para manejar un determinado tipo de carga de trabajo, tenga en cuenta los requisitos de almacenamiento de la carga de trabajo.

¿Qué tamaño debe tener la base de datos para manejar sus requisitos de datos?  
¿Necesita almacenar gigabytes? ¿Terabytes? ¿Petabytes?

Las diferentes arquitecturas de bases de datos admiten diferentes capacidades máximas de datos. Algunos diseños de bases de datos son ideales para aplicaciones tradicionales, mientras que otros son ideales para el almacenamiento en caché o la administración de sesiones. Otros son ideales para Internet de las cosas (IoT) o aplicaciones de big data.

Es fundamental comprender los requisitos de almacenamiento total para elegir una base de datos.

## Consideraciones de la base de datos: tamaño y tipo de objetos



Escalabilidad



Requisitos de almacenamiento totales

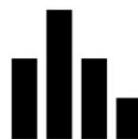


**Tamaño y tipo de objeto**



Durabilidad

¿Es necesario almacenar estructuras  
de datos simples, objetos de datos  
grandes o ambos?



En tercer lugar, cuando elija una base de datos para sus cargas de trabajo específicas, tenga en cuenta el tamaño y el tipo de objetos que debe almacenar.

¿Es necesario almacenar estructuras de datos simples, objetos de datos grandes o ambos?

## Consideraciones de la base de datos: durabilidad



Escalabilidad



Requisitos de almacenamiento totales



Tamaño y tipo de objeto



Durabilidad

¿Qué nivel de durabilidad, disponibilidad y capacidad de recuperación de datos se requiere?

¿Se aplican obligaciones normativas?



Por último, como cuarta consideración para la elección de su base de datos, piense en los requisitos de durabilidad de los datos que almacenará.

La *durabilidad de los datos* se refiere a la garantía de que sus datos no se perderán, y la *disponibilidad de los datos* se refiere a su capacidad para acceder a los datos cuando lo deseé. ¿Qué nivel de durabilidad y disponibilidad de datos necesita? Si los datos que almacenará son absolutamente fundamentales para su empresa, debe elegir una solución de base de datos que almacene varias copias redundantes de los datos en varias ubicaciones físicas separadas de forma geográfica. Por lo general, esta solución generará un aumento del costo, por lo que es importante equilibrar sus necesidades empresariales con las consideraciones de costos.

Otra consideración importante es saber si la residencia de datos o las *obligaciones normativas* se aplican a sus datos. Por ejemplo, ¿existen leyes regionales de privacidad de los datos que deba cumplir? Si es así, elija una solución de base de datos que admita la conformidad.

## Tipos de bases de datos



Ahora que revisó las consideraciones clave, considere las dos categorías de opciones de base de datos disponibles:

### Relacional

Ejemplos tradicionales

Microsoft SQL Server  
Base de datos de Oracle  
MySQL

### No relacional

Ejemplos tradicionales

MongoDB  
Cassandra  
Redis

Puede elegir entre muchos tipos de bases de datos, muchas de las cuales están personalizadas. Sin embargo, los tipos de base de datos suelen clasificarse en una de las dos categorías generales: relacional o no relacional.

Los *sistemas de bases de datos relationales* son el tipo de base de datos más familiar para la mayoría de las personas. Los ejemplos tradicionales incluyen Microsoft SQL Server, Oracle Database y MySQL.

Las bases de datos no relationales se desarrollaron más recientemente, pero han existido desde hace algunas décadas. Cumplen un rol esencial en el panorama informático moderno. Algunos ejemplos incluyen MongoDB, Cassandra y Redis.

## Tipo de base de datos relacional

**aws academy**

**Beneficios:**

- Facilidad de uso
- Integridad de los datos
- Almacenamiento de datos reducido
- Lenguaje común (lenguaje de consulta estructurada o SQL)



Sistema de administración de bases de datos relacionales (RDBMS)

La base de datos relacional es ideal cuando sucede lo siguiente:

- Necesita reglas estrictas de esquema, conformidad con ACID y cumplimiento de la calidad de los datos.
- No necesita una capacidad extrema de lectura o escritura.
- No necesita un rendimiento extremo.
  - Un RDBMS puede ser la mejor solución de menor esfuerzo



Las bases de datos relacionales a veces se denominan *sistemas de administración de bases de datos relacionales* (RDBMS). Siguen siendo la categoría de base de datos más popular y más utilizada en el mundo. En un [estudio de clasificación de DB-Engines](#) de 2019, se descubrió que representan más del 75 % de la puntuación de popularidad en todas las categorías. Los cuatro primeros fueron Oracle, MySQL, Microsoft SQL Server y PostgreSQL; todas son bases de datos relacionales.

Las bases de datos relacionales mantienen su popularidad por muchas razones. Estas razones incluyen su *facilidad de uso*, controles de integridad de los datos, excelencia en la reducción del almacenamiento general de datos y su compatibilidad con el lenguaje de consulta estructurada (SQL). SQL es un lenguaje popular para consultar o interactuar con datos estructurados almacenados en un RDBMS, y es compatible con la mayoría de los proveedores.

Si su caso de uso se adapta bien a las *reglas estrictas de esquema*, donde la estructura del esquema está bien definida y no necesita cambiar a menudo, una base de datos relacional sería una buena opción. Las transacciones de bases de datos relacionales también son compatibles con *ACID*, lo que significa que garantizan la integridad de los datos proporcionando transacciones *atómicas, consistentes, aisladas y duraderas*. Sin embargo, si su aplicación necesita una

capacidad extrema de lectura o escritura, es posible que una base de datos relacional no sea la opción adecuada. Finalmente, el rendimiento extremo es una característica de las bases de datos relacionales cuando las compara con otras opciones. Sin embargo, una base de datos relacional puede ser la mejor solución de mínimo esfuerzo para muchos casos de uso.

# Tipo de base de datos no relacional

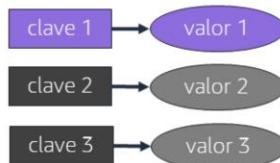


## Beneficios

- Flexibilidad
- Escalabilidad
- Alto rendimiento
- API altamente funcionales

## Modelos de ejemplo

### Clave-valor



La base de datos no relacional es ideal cuando sucede lo siguiente:

- La base de datos debe escalarse **de forma horizontal** para manejar un volumen de datos masivo.
- Los datos no se ajustan bien a los esquemas tradicionales.
- Las tasas de lectura o escritura superan las que se pueden sustentar económicamente por medio de un RDBMS tradicional.



Las bases de datos no relacionales a veces se denominan *bases de datos NoSQL*. Son diferentes de las bases de datos relacionales porque están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos no relacionales son ampliamente apreciadas por su *flexibilidad, escalabilidad, alto rendimiento y API altamente funcionales*.

Usan una variedad de modelos de datos, que incluyen clave-valor, gráficos, documentos, datos en memoria y búsqueda. Los esquemas NoSQL son dinámicos. Por ejemplo, una fila no necesita contener datos para cada columna. Además, una base de datos NoSQL puede *escalarse de forma horizontal* mediante el aumento del número de servidores en los que se ejecuta.

Las bases de datos no relacionales tienen *esquemas flexibles*, en los que cada objeto puede tener una estructura diferente. Pueden almacenar datos estructurados, como registros de bases de datos relacionales; datos semiestructurados, como documentos de notación de objetos JavaScript (JSON); y datos no estructurados, como archivos de fotos o mensajes de correo electrónico. Las bases de datos no relacionales funcionan bien cuando los datos pueden tener inconsistencias estructurales.

Finalmente, las bases de datos no relacionales están optimizadas para modelos de datos específicos y patrones de acceso que permiten un *mayor rendimiento* en lugar de intentar lograr la funcionalidad de las bases de datos relacionales.

# Opciones de base de datos de Amazon

aws academy

Existen más opciones de bases de datos; estas opciones son ejemplos comunes

Bases de datos relacionales



Amazon RDS

Bases de datos no relacionales



Amazon Redshift



Amazon Aurora



Amazon DynamoDB



Amazon ElastiCache



Amazon Neptune

↑ Céntrese en este módulo ↑

Ahora que aprendió acerca de las diferencias entre las bases de datos *relacionales* y las *no relacionales*, considere algunos ejemplos de ofertas de bases de datos de Amazon que se adaptan a esas dos categorías.

Entre las ofertas de bases de datos *relacionales* de AWS más utilizadas se encuentran Amazon RDS, Amazon Redshift y Amazon Aurora.

Del mismo modo, algunas de las bases de datos *no relacionales* más utilizadas son Amazon DynamoDB, Amazon ElastiCache y Amazon Neptune.

En este módulo, analizará Amazon RDS y Amazon DynamoDB en más profundidad. Para obtener una lista completa de los servicios de bases de datos disponibles en AWS y sus casos de uso más comunes, consulte la [página de información general de las bases de datos de AWS](#).

## Aprendizajes clave de la Sección 2



The slide features a large, ornate key lying on a teal-colored wooden surface. A small white rectangular tag is tied to the key's ring, with the word "Takeaway" written on it in a cursive font.

aws academy

- Cuando elija una base de datos, tenga en cuenta la **escalabilidad**, los **requisitos de almacenamiento**, el **tipo y tamaño de los objetos** que se almacenarán y los **requisitos de durabilidad**.
- **Las bases de datos relacionales** tienen reglas estrictas de esquema, proporcionan integridad de los datos y admiten SQL.
- **Las bases de datos no relacionales** se escalan de forma horizontal, proporcionan mayor escalabilidad y flexibilidad, y funcionan bien para datos semiestructurados y no estructurados.

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

- Cuando elija una base de datos, tenga en cuenta la escalabilidad, los requisitos de almacenamiento, el tipo y tamaño de los objetos que se almacenarán y los requisitos de durabilidad.
- Las bases de datos relacionales tienen reglas estrictas de esquema, proporcionan integridad de los datos y admiten SQL.
- Las bases de datos no relacionales se escalan de forma horizontal, proporcionan mayor escalabilidad y flexibilidad, y funcionan bien para datos semiestructurados y no estructurados.

Módulo 5: Incorporación de una capa de base de datos

Sección 3: Amazon RDS



Presentación de la Sección 3: Amazon RDS.

## Amazon RDS



Relacional

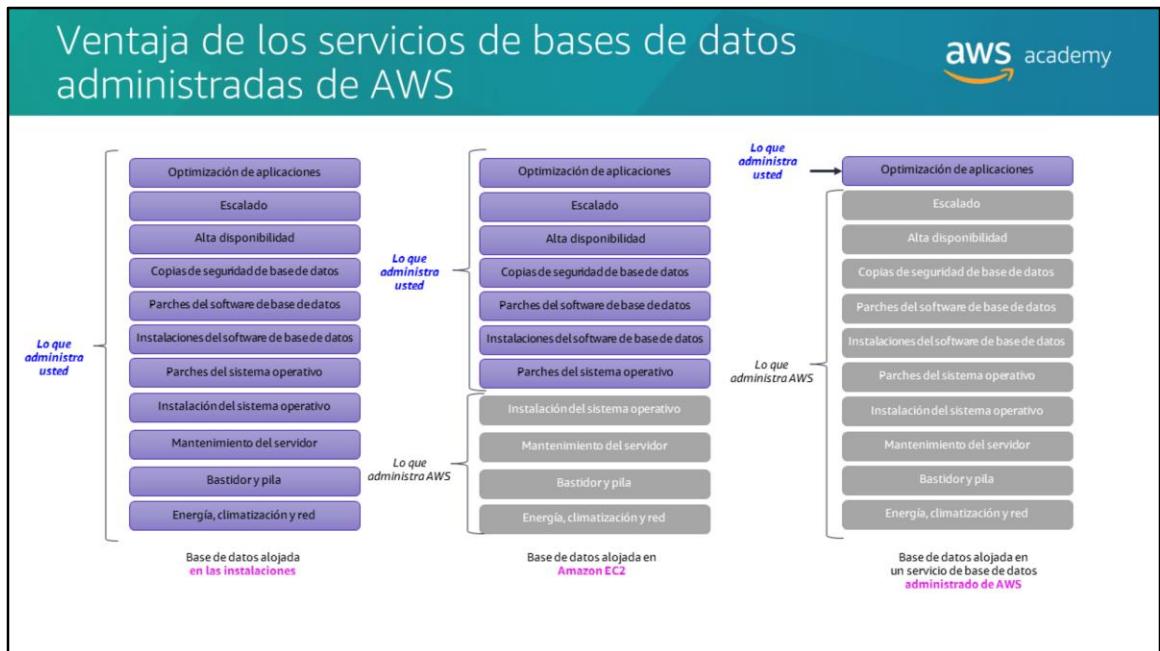


Amazon  
RDS

Amazon RDS es un servicio de base de datos relacional completamente administrada.



Amazon RDS es un servicio de base de datos relacional completamente administrada que crea y opera una base de datos relacional en la nube. Sin embargo, antes de obtener más información acerca de Amazon RDS, revisará las ventajas de Amazon RDS como servicio de base de datos *administrada*.



Ahora considerará algunas de las ventajas de utilizar los servicios de bases de datos administradas de AWS.

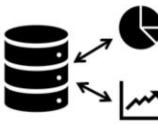
En el caso de uso de la izquierda, aloja la base de datos en el centro de datos *en las instalaciones*. En este caso, usted es responsable de todo. Debe alimentar los bastidores de los servidores físicos, mantener los sistemas operativos de los servidores donde se ejecutan las bases de datos y realizar instalaciones y aplicar parches de software de base de datos. También debe realizar copias de seguridad, configurar soluciones de escalado y alta disponibilidad, y optimizar las aplicaciones que utilizan la base de datos.

En el caso de uso del centro, instalará una *base de datos en una o más instancias de Amazon EC2*. En este caso, AWS gestiona el mantenimiento del entorno del centro de datos físico y el sistema operativo está preinstalado en la instancia EC2 que inicie. Sin embargo, usted sigue siendo responsable de cada capa de configuración por encima de la instalación del sistema operativo, lo que significa que debe administrar muchos recursos manualmente.

La *oferta de base de datos administrada de AWS* se encuentra en el caso de uso de la derecha. Estas soluciones proporcionan alta disponibilidad, escalabilidad y copias de seguridad de la base de datos como opciones integradas que puede configurar. AWS es responsable de gestionar las tareas comunes y repetitivas de administración de las bases de datos. Usted solo es responsable de optimizar la aplicación y asegurarse de que la capa de base de datos funcione de manera eficaz para la aplicación.

## Características de Amazon RDS



			
<b>Patrón de acceso</b> Transaccional, análisis ligero	<b>Volumen de datos</b> Rango de pocos TB	<b>Eficacia</b> Rendimiento de medio a alto, baja latencia	<b>Casos de uso empresariales</b> Transaccional OLAP

Amazon Relational Database Service (Amazon RDS) es un servicio de base de datos relacional completamente administrada.

Amazon RDS se utiliza normalmente cuando el patrón de acceso es transaccional o para análisis ligeros.

Como base de datos relacional, el tamaño ideal de los datos oscila hasta el rango bajo de terabytes. Según aumenten los requisitos, puede aprovisionar almacenamiento adicional. El motor Amazon Aurora aumentará automáticamente el tamaño del volumen de la base de datos a medida que aumenten las necesidades de almacenamiento de la base de datos. Se escala hasta un máximo de 64 TB, o el máximo que usted defina. Los motores MySQL, MariaDB, Oracle y PostgreSQL le permiten escalar hasta 32 TB de almacenamiento.

Microsoft SQL Server admite hasta 16 TB. El escalado del almacenamiento es dinámico, sin tiempo de inactividad.

En términos de *rendimiento*, Amazon RDS ofrece dos opciones. En primer lugar, ofrece la opción de almacenamiento respaldado por una unidad de estado sólido (SSD) de uso general. La opción de SSD ofrece una base de referencia consistente de 3 IOPS por GB aprovisionado, y puede ampliar su uso mediante ráfagas de hasta

3000 IOPS por encima de la base de referencia. Este tipo de almacenamiento es apto para una gama de cargas de trabajo de base de datos. También hay una opción de almacenamiento en SSD con IOPS provisionadas, que funciona bien para cargas de trabajo de bases de datos transaccionales con uso intensivo de operaciones de E/S.

## Amazon RDS: usos y tipos de base de datos



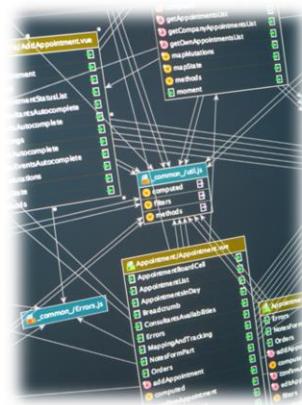
Amazon  
RDS

Funciona bien para aplicaciones que cumplen con lo siguiente:

- Tienen datos más complejos
- Necesitan combinar y unir conjuntos de datos.
- Necesitan reglas de sintaxis aplicadas.

Se admiten seis tipos de bases de datos:

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>• Microsoft SQL Server</li><li>• Oracle</li><li>• MySQL</li></ul> | <ul style="list-style-type: none"><li>• PostgreSQL</li><li>• Aurora</li><li>• MariaDB</li></ul> |
|---|---|



Como oferta de bases de datos relacionales, Amazon RDS es una buena opción para aplicaciones que tienen datos complejos y bien estructurados. Amazon RDS es una buena opción si las cargas de trabajo deben combinar y unir conjuntos de datos con frecuencia, y deben tener reglas de sintaxis que se apliquen estrictamente. Por ejemplo, Amazon RDS se utiliza con frecuencia para respaldar aplicaciones tradicionales, aplicaciones de planificación de recursos empresariales (ERP), aplicaciones de gestión de relaciones con clientes (CRM) y aplicaciones de comercio electrónico.

Amazon RDS está disponible con seis motores de base de datos entre los que puede elegir, incluidos Microsoft SQL Server, Oracle, MySQL, PostgreSQL, Amazon Aurora y MariaDB.

Tamaño de la instancia de base de datos			
	Familia T	Familia M	Familia R
Tipo	Instancias ampliables	Instancias de uso general	Instancias con optimización de memoria
Tamaño	1 vCPU/1 GB de RAM a 8 vCPU/32 GB de RAM	2 vCPU/8 GB de RAM a 96 vCPU/384 GB de RAM	2 vCPU/16 GB de RAM a 96 vCPU/768 GB de RAM
Redes	Rendimiento moderado	Alto rendimiento	Alto rendimiento
Carga de trabajo ideal	Menor o variable	Mucho consumo de CPU	Recuentos altos de conexiones con uso intensivo de consultas
Aspectos destacados	T3 puede ampliarse por encima de la referencia por un cargo adicional	M5 ofrece hasta 96 vCPU	R5 ofrece hasta 96 vCPU/768 GiB de RAM

Todos los tipos de bases de datos de Amazon RDS se ejecutan en un servidor. La excepción es Aurora, que puede ejecutarse como una opción sin servidor. Amazon RDS está disponible en varios tipos de instancias de base de datos, optimizadas para diferentes tipos de cargas de trabajo.

La familia T de tipos de instancia proporciona un nivel base de rendimiento de la CPU con la capacidad para ampliar su uso mediante ráfagas en cualquier momento durante el tiempo que resulte necesario. Por ejemplo, las instancias T3 ofrecen un equilibrio entre recursos informáticos, de memoria y de red, y están diseñadas para cargas de trabajo de bases de datos con un uso moderado de la CPU que experimentan picos temporales de uso.

La familia M de instancias es otra opción de uso general. Sin embargo, la familia M ofrece opciones adicionales para cargas de trabajo intensivas con uso intensivo de la CPU. Las instancias M son una buena opción para bases de datos pequeñas y medianas para aplicaciones empresariales o de código abierto.

Por último, las instancias de la familia R están optimizadas para cargas de trabajo de bases de datos con uso intensivo de la memoria.

Algunos motores de base de datos admiten clases de instancias de bases de datos adicionales. Para obtener más información, consulte [Instancias de base de datos de Amazon RDS](#) y, en concreto, consulte la página de documentación de AWS [Elegir la clase de instancias de bases de datos](#) para obtener detalles de la clase de instancia.



Considere un ejemplo de caso de uso en el que Amazon RDS se utiliza como parte de una solución implementada para un desafío empresarial.

En este ejemplo, una empresa desea obtener información significativa a partir de algunos de sus datos empresariales. Los usuarios cargan los datos en un bucket de Amazon Simple Storage Service (Amazon S3). A continuación, se utiliza un clúster de instancias de Amazon EC2 para transformar los datos. Esto puede ocurrir como un procesamiento por lotes a un intervalo regular. Luego, los datos transformados se almacenan en un bucket de S3 diferente. Sin embargo, algunos de los datos transformados también se insertan en una instancia de RDS, que se etiqueta como *Almacén de datos*. A continuación, se puede consultar esta instancia de RDS utilizando software de análisis o un servicio de análisis de AWS que puede realizar consultas SQL para acceder a los datos para obtener información empresarial.

Módulo 5 -  
Laboratorio guiado:  
Creación de una  
base de datos  
de Amazon RDS



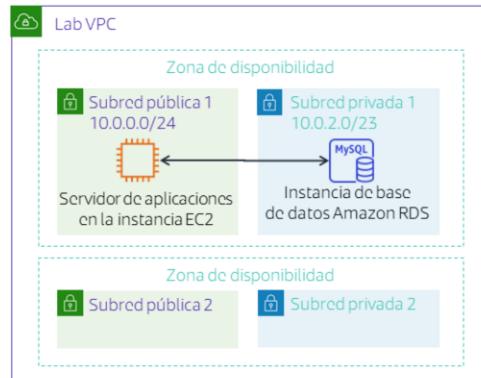
aws academy

Ahora completará el Módulo 5 - Laboratorio guiado: Creación de una base de datos de Amazon RDS.

## Laboratorio guiado: Tareas



1. Creación de una base de datos de Amazon RDS
2. Configuración de la comunicación de aplicaciones web con una instancia de base de datos



En este laboratorio guiado, realizará las siguientes tareas:

1. Creación de una base de datos de Amazon RDS
2. Configuración de la comunicación de aplicaciones web con una instancia de base de datos



A circular icon with a power symbol and the text "~ 20 minutos".

The AWS Academy logo.

Comenzar Módulo 5 -  
Laboratorio guiado:  
Creación de una base  
de datos de  
Amazon RDS

Llegó la hora de iniciar el laboratorio guiado.



## Análisis del laboratorio guiado: aprendizajes clave



El instructor podría mediar una conversación sobre los aprendizajes clave de este laboratorio guiado, una vez que haya terminado.

# Implementación Multi-AZ para alta disponibilidad

**aws academy**

**Beneficios**

- Durabilidad mejorada
- Mayor disponibilidad
- La comutación por error al modo de espera se produce automáticamente

**Condiciones de la comutación por error automatizada**

- Pérdida de disponibilidad en la zona de disponibilidad principal
- Pérdida de conectividad de red con la instancia principal
- Error de unidad informática en la instancia principal
- Error de almacenamiento en instancia principal

Dirige el tráfico hacia las aplicaciones

Amazon Route 53

Región

Zona de disponibilidad 1

Zona de disponibilidad 2

Grupo de Auto Scaling

Instancias de la aplicación

Principal

Secundario

Replicación síncrona

Amazon RDS

Volumen de EBS

Amazon RDS ofrece alta disponibilidad mediante la implementación de un enfoque Multi-AZ. Amazon RDS aprovisiona y mantiene automáticamente una instancia en espera síncrona dentro de una zona de disponibilidad diferente. La instancia de base de datos principal se replica de forma síncrona en distintas zonas de disponibilidad en una instancia en espera para proporcionar redundancia de datos, eliminar las congelaciones de E/S y minimizar los picos de latencia durante las copias de seguridad del sistema.

La ejecución de una instancia de base de datos con alta disponibilidad puede mejorar la disponibilidad durante el mantenimiento planificado del sistema. Ayuda a proteger las bases de datos contra errores en la instancia de base de datos e interrupciones en la zona de disponibilidad. También ofrece disponibilidad y durabilidad mejoradas para las instancias de base de datos, lo que las convierte en una buena opción para cargas de trabajo de bases de datos de producción.

Si ocurre un error de infraestructura, Amazon RDS realiza una comutación por error automática a la instancia en espera. A continuación, puede reanudar las operaciones de la base de datos cuando se complete la comutación por error. El punto de enlace de la instancia de base de datos seguirá siendo el mismo tras la comutación por error, por lo tanto, su aplicación podrá reanudar las operaciones de la base de datos sin necesidad de realizar una intervención administrativa manual.

Las condiciones de conmutación por error incluyen la pérdida de disponibilidad en la zona de disponibilidad principal, la pérdida de conectividad de red con la base de datos principal, un error de la unidad informática en la instancia principal o un error de almacenamiento en la instancia principal.

## Réplicas de lectura para el rendimiento



### Beneficios

- Rendimiento mejorado
- Mayor disponibilidad
- Diseñado para garantizar la seguridad

### Compatible con

- MySQL
- MariaDB
- PostgreSQL
- Oracle

### Límites

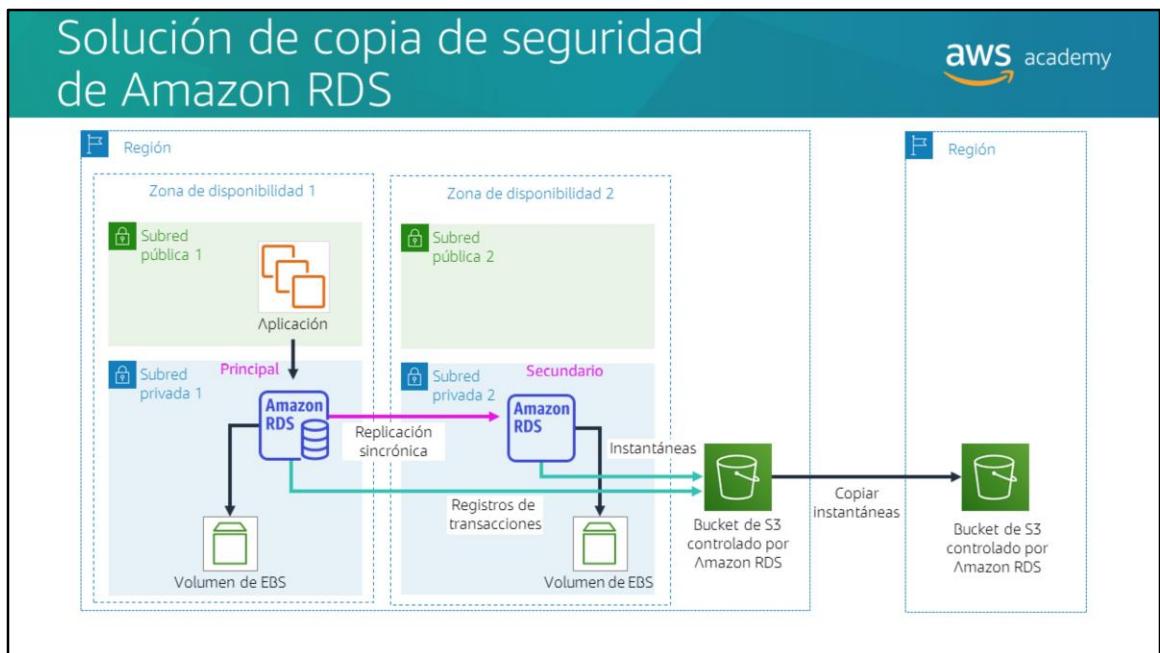
- Cinco réplicas de lectura por instancia principal
- Para obtener una consistencia estricta de lectura después de escritura, lea desde la instancia principal



Amazon RDS le permite utilizar una instancia de base de datos de origen para crear un tipo especial de instancia de base de datos, que se denomina *réplica de lectura*. Las actualizaciones que se hacen en la instancia de base de datos de origen se copian de forma asincrónica en la réplica de lectura.

El uso de réplicas de lectura mejora *el rendimiento*. Por ejemplo, puede reducir la carga de la instancia de base de datos de origen mediante el direccionamiento de las consultas de lectura de sus aplicaciones a la réplica de lectura. Las réplicas de lectura también aumentan la *disponibilidad*. Para las cargas de trabajo de base de datos que realizan un uso intensivo de las lecturas, puede escalar de manera elástica más allá de los límites de capacidad de una única instancia de base de datos. Las réplicas de lectura también se pueden promover manualmente para que se transformen en instancias de base de datos independientes, cuando sea necesario.

Las réplicas de lectura están disponibles en Amazon RDS for MySQL, MariaDB, PostgreSQL y Oracle. Cada uno de estos motores de base de datos permite hasta un máximo de cinco réplicas de lectura por base de datos principal. Si necesita una consistencia estricta de lectura después de escritura (es decir, lo que *lo que lee es siempre lo que acaba de escribir*, incluso si lo lee inmediatamente), entonces debe leer desde la instancia de base de datos principal. De lo contrario, puede distribuir la carga y leer desde una réplica de lectura.



Ahora, considere un ejemplo de arquitectura para una solución de copia de seguridad basada en Amazon RDS.

Aquí, las instantáneas y los registros de transacciones de la base de datos de RDS se almacenan en un bucket de S3 controlado por Amazon RDS. Con Amazon RDS, también puede especificar que copie la base de datos automatizada o manual en una segunda región de AWS. Además, puede copiar las instantáneas en cuentas separadas de AWS.

Para obtener más información, consulte [Copiar una instantánea](#) en la documentación de AWS.

Demostración:  
Réplicas de lectura  
y copias de seguridad  
automatizadas de  
Amazon RDS



aws academy



Ahora, el instructor puede elegir demostrar la configuración de copias de seguridad automatizadas y réplicas de lectura en Amazon RDS mediante la consola de administración de AWS.

## Amazon Aurora



Amazon Aurora es un motor de base de datos relacional completamente administrado compatible con MySQL y PostgreSQL.



Amazon Aurora

- Se usa para el procesamiento de transacciones en línea (OLTP).
- Ofrece hasta cinco veces más rendimiento que MySQL\*.
- Ofrece hasta tres veces más rendimiento que PostgreSQL\*.
- Replica datos de seis maneras en tres zonas de disponibilidad.
- Requiere pocos cambios en su aplicación existente.

\* Los detalles del punto de referencia están disponibles para [MySQL](#) y [PostgreSQL](#).

Amazon Aurora es una de las opciones de motor de base de datos para Amazon RDS.

Aurora es una base de datos relacional compatible con MySQL y PostgreSQL creada para la nube. Combina el rendimiento y la disponibilidad de las bases de datos empresariales tradicionales con la simplicidad y la rentabilidad de las bases de datos de código abierto.

Es hasta cinco veces más rápida que las bases de datos MySQL estándar y tres veces más rápida que las bases de datos PostgreSQL estándar. Ofrece la seguridad, la disponibilidad y la fiabilidad de las bases de datos de nivel comercial por una décima parte del costo. Las instrucciones detalladas sobre esta referencia y cómo replicarla se proporcionan en la [Guía de referencia de rendimiento de Amazon Aurora MySQL](#) y en la [Guía de referencia de rendimiento de Amazon Aurora PostgreSQL](#).

Aurora ofrece un sistema de almacenamiento distribuido, tolerante a errores y de recuperación automática que ajusta su escala automáticamente hasta 64 TB por instancia de base de datos. Aurora suministra alto rendimiento y disponibilidad con hasta 15 réplicas de lectura de baja latencia, recuperación a un momento dado, generación de copias de seguridad continua en Amazon S3 y replicación en tres zonas de disponibilidad.

Aurora se utiliza con frecuencia para el *procesamiento de transacciones en línea* (OLTP). Los sistemas de OLTP deben poder manejar un gran volumen de usuarios simultáneos y poder ejecutar solicitudes de inserción y actualización. Un ejemplo de OLTP común es un sistema de entrada de pedidos. Con frecuencia, el OLTP se contrasta con el *procesamiento de análisis en línea* (OLAP), que se caracteriza por un volumen menor de consultas más complejas.

## Amazon Redshift



Amazon Redshift es un servicio de almacenamiento de datos.

- Se utiliza para el procesamiento de análisis en línea (OLAP).
- Almacena conjuntos muy grandes de datos.
  - Almacena datos muy estructurados y de acceso frecuente en Amazon Redshift.
  - También puede almacenar exabytes de datos estructurados, semiestructurados y no estructurados en Amazon S3.

Amazon Redshift es una oferta diferente de base de datos relacional de AWS. No se ejecuta en Amazon RDS.

Amazon Redshift ofrece un almacenamiento de datos a escala de petabytes y análisis de lagos de datos. Amazon Redshift suele utilizarse para almacenar datos muy estructurados a los que se accede con frecuencia. Amazon Redshift también puede acceder a los datos directamente en Amazon S3, por lo que también puede mantener exabytes de datos estructurados, semiestructurados y no estructurados en Amazon S3.

Amazon Redshift ofrece un rendimiento constante y rápido, incluso con miles de consultas simultáneas. Funciona de forma consistente independientemente de si está consultando datos en el almacenamiento de datos de Amazon Redshift o en el lago de datos de Amazon S3. Puede consultar formatos de archivo abiertos como Parquet, JSON, Avro, CSV. También puede consultar Amazon S3 directamente mediante SQL.



Aprendizajes clave de la Sección 3

**Takeaway**

The slide features a large key icon on the left and a list of learning objectives on the right.

**aws academy**

- Los servicios de bases de datos administradas de AWS manejan las tareas de administración para que pueda centrarse en sus aplicaciones.
- Amazon RDS admite Microsoft SQL Server, Oracle, MySQL, PostgreSQL, Aurora y MariaDB.
- Las implementaciones Multi-AZ de Amazon RDS ofrecen alta disponibilidad con conmutación por error automática.
- Puede tener hasta cinco réplicas de lectura por base de datos principal para mejorar el rendimiento de Amazon RDS.
- Amazon Aurora es un motor de base de datos relacional completamente administrada compatible con MySQL y PostgreSQL.
- Amazon Redshift es una oferta de base de datos relacional para el almacenamiento de datos.

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

- Los servicios de bases de datos administradas de AWS manejan las tareas de administración para que pueda centrarse en sus aplicaciones
- Amazon RDS admite Microsoft SQL Server, Oracle, MySQL, PostgreSQL, Aurora y MariaDB
- Las implementaciones Multi-AZ de Amazon RDS ofrecen alta disponibilidad con conmutación por error automática
- Puede tener hasta cinco réplicas de lectura por base de datos principal para mejorar el rendimiento
- Amazon Aurora es un motor de base de datos relacional completamente administrada compatible con MySQL y PostgreSQL
- Amazon Redshift es una oferta de base de datos relacional de almacenamiento de datos

Módulo 5: Incorporación de una capa de base de datos

Sección 4: Amazon DynamoDB



Presentación de la Sección 4: Amazon DynamoDB.

# Amazon DynamoDB

aws academy



Amazon  
DynamoDB

Un servicio de base de datos **no relacional** completamente administrada **de clave-valor y documentos**.

	Eficacia a cualquier escala	
	Sin servidor	
	Lista para el uso empresarial	
Capacidad extrema de escalado horizontal	Programación dirigida por eventos (informática sin servidor)	Cifrado, controles de acceso, copias de seguridad

Amazon DynamoDB es un servicio de base de datos NoSQL no relacional completamente administrada. Proporciona un rendimiento rápido y previsible, así como una escalabilidad sin dificultades.

Entre los beneficios principales se incluyen:

- Rendimiento a gran escala
  - Ofrece tiempos de respuesta uniformes en menos de diez milisegundos.
  - Puede crear aplicaciones con capacidad de procesamiento prácticamente ilimitada.
- Sin servidor
  - No hay servidores para aprovisionar, aplicar parches o administrar.
  - No hay software para instalar, mantener u operar.
- Lista para el uso empresarial
  - Admite transacciones ACID.
  - Cifra todos los datos en reposo de forma predeterminada.
  - Está disponible la replicación en varias regiones (tablas globales).
  - Proporciona control de acceso e identidad detallado.
  - Puede realizar copias de seguridad completas de los datos sin impacto en el rendimiento.

## Características de Amazon DynamoDB



Amazon  
DynamoDB

Funciona bien para aplicaciones que cumplen con lo siguiente:

- Tienen grandes volúmenes de datos simples (rango alto de TB).
- Deben escalarse rápidamente.
- No necesitan combinaciones complejas.
- Requieren un rendimiento ultraalto y baja latencia.

Características clave

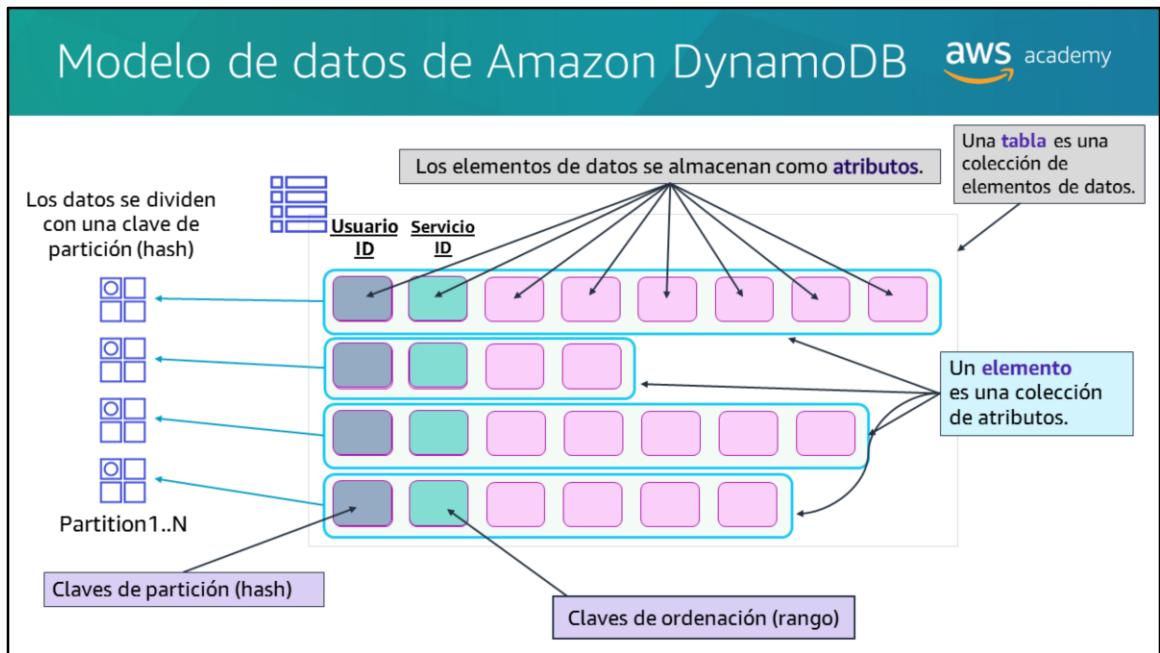
- Tablas NoSQL
- Los elementos pueden tener atributos diferentes
- Almacenamiento de caché en memoria
- Compatibilidad con picos de más de 20 millones de solicitudes por segundo



DynamoDB funciona bien para aplicaciones que manejan un gran volumen de datos y deben escalarse rápidamente. También proporciona un rendimiento ultraalto y baja latencia, por lo que es una buena opción para aplicaciones de videojuegos, de tecnología publicitaria, móviles y otras aplicaciones que tienen estos requisitos. Sin embargo, una base de datos relacional es probablemente una mejor opción si las cargas de trabajo requieren combinaciones complejas.

Las tablas de DynamoDB no presentan esquemas fijos, por lo que cada elemento puede tener un número distinto de atributos.

Dado que DynamoDB ofrece almacenamiento de caché en memoria y puede manejar más de 20 millones de solicitudes por segundo, se utiliza con frecuencia para aplicaciones que deben mantener el estado de la sesión, para aplicaciones web sin servidor y para microservicios.



Ahora, considere el modelo de datos de Amazon DynamoDB. A diferencia de las bases de datos relacionales, y debido a que es una base de datos NoSQL, DynamoDB no aplica esquemas estrictos.

Una tabla de DynamoDB contiene *elementos*. Un elemento tiene una cantidad variable de *atributos*. Cada atributo está compuesto por un *par clave-valor*.

DynamoDB admite operaciones clave-valor GET o PUT que utilizan una clave principal definida por el usuario (conocida como “clave de partición” o “clave hash”). La clave principal es el único atributo necesario para los elementos de una tabla, ya que identifica cada uno de ellos. La clave principal se especifica cuando se crea una tabla. Asimismo, DynamoDB permite realizar consultas flexibles a los atributos clave no principales mediante índices secundarios globales e índices secundarios locales.

Las particiones automáticas se producen a medida que crece el conjunto de datos y aumenta la capacidad aprovisionada.

Considere un caso de ejemplo en el que crea una tabla de Amazon DynamoDB para conservar la información de los pedidos. En el ejemplo, las entradas de la tabla siempre tendrán un *ID de usuario* y un *ID de pedido*. Sin embargo, algunos de los otros detalles sobre un pedido, como la dirección de correo electrónico del usuario,

solo se pueden recopilar para algunos pedidos y no para otros. A medida que se agregan elementos adicionales a la tabla, puede seguir recopilando nuevos atributos. Estos atributos pueden incluir aquellos que no esperaba cuando creó la tabla. Esta flexibilidad es lo que hace que DynamoDB sea una buena opción para datos no estructurados, semiestructurados y estructurados que cambian rápidamente.



De forma predeterminada, Amazon DynamoDB replica los datos en varias zonas de disponibilidad en una región única. Sin embargo, es posible que haya casos en los que desee replicar los datos en varias regiones.

Las tablas globales de Amazon DynamoDB proporcionan una solución completamente administrada para implementar una base de datos en varias regiones y con varios nodos maestros. No necesita crear y mantener su propia solución de replicación. Cuando se crea una tabla global, debe especificar las regiones de AWS en las que desea que la tabla esté disponible. DynamoDB realiza todas las tareas necesarias para crear tablas idénticas en estas regiones. DynamoDB también propaga los cambios que se vayan realizando en los datos en todas las tablas.

Por poner un ejemplo de un caso de uso de una tabla global, imagine que tiene una gran cartera de clientes repartidos en tres áreas geográficas: la costa este de Estados Unidos, la costa oeste de Estados Unidos y Europa Occidental. Los clientes tienen que actualizar su información de perfil mientras usan la aplicación.

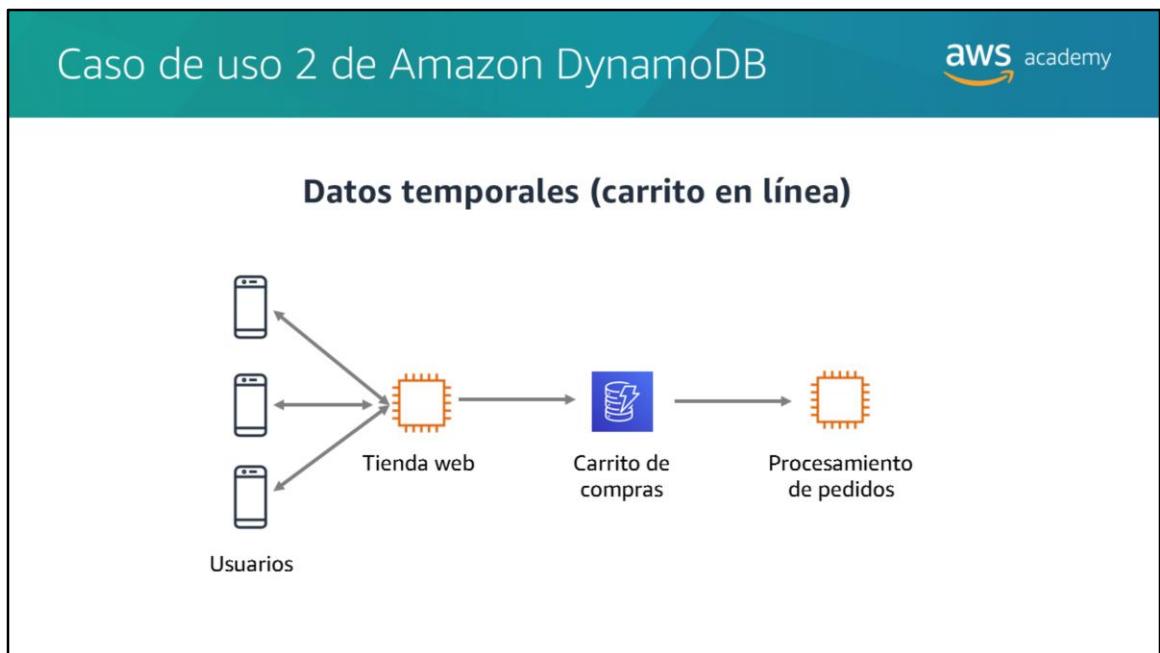
Para manejar este requisito empresarial, puede crear una tabla global que se componga de tres tablas CustomerProfiles específicas de cada región. Luego, DynamoDB replicará automáticamente los cambios en los datos entre esas tablas.

Los cambios en los datos de CustomerProfiles en una región se propagarían sin problemas a las otras regiones. Asimismo, aunque una de las regiones de AWS no esté disponible durante un tiempo, sus clientes todavía pueden obtener acceso a los mismos datos de CustomerProfiles desde otras regiones.



Las empresas de videojuegos utilizan Amazon DynamoDB en todas las partes de sus plataformas de videojuegos, incluidos el estado del juego, los datos de los jugadores, el historial de sesiones y los marcadores. Puede escalar con fiabilidad a millones de solicitudes y usuarios simultáneos y, al mismo tiempo, garantizar una latencia baja de manera constante.

En el caso de uso de ejemplo, se mantiene un marcador de los juegos en Amazon DynamoDB. Los jugadores que están conectados activamente a los servidores del juego podrían ver parte de esta información presentada de la manera en que la interfaz de usuario (IU) del videojuego se la brinde. El marcador GameScores se actualiza constantemente cuando se alcanzan nuevas puntuaciones máximas y cuando cambia el récord de victorias o derrotas de cada uno de los mejores jugadores. Como ejemplo, considere Electronic Arts (EA), una gran empresa de videojuegos con más de 300 millones de jugadores registrados en todo el mundo. Para EA, una simultaneidad alta puede significar más de 100 000 solicitudes por segundo y millones de usuarios activos por día. Consulte [Casos de uso de videojuegos de DynamoDB](#) para obtener más detalles.



Amazon DynamoDB se desarrolló porque los clientes necesitaban una base de datos de clave-valor escalable de confianza para impulsar las operaciones esenciales de comercio electrónico de Amazon, como el carrito de compras.

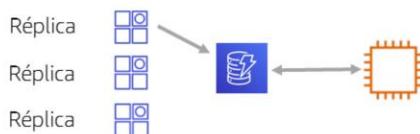
En 2012, el director de tecnología de Amazon, Werner Vogels, explicó los requisitos empresariales del carrito de compras de Amazon en una [publicación de blog](#) cuando anunció el lanzamiento de Amazon DynamoDB. “Esta base de datos no relacional, o NoSQL, estaba destinada a los casos de uso... como el carrito de compras... Cualquier tiempo de inactividad o degradación del rendimiento... tiene un impacto financiero inmediato y los requisitos de tolerancia a errores y rendimiento para sus sistemas de datos son muy estrictos. Estos servicios también requieren la capacidad de escalar la infraestructura gradualmente para acomodar el crecimiento de las tasas de solicitudes o los tamaños de los conjuntos de datos. Otro requisito importante... era la predictibilidad”.

El conjunto de características de DynamoDB sigue evolucionando. Sin embargo, los objetivos principales que inspiraron el desarrollo del servicio siguen siendo relevantes en la actualidad para las organizaciones que desean implementar una solución de procesamiento para los pedidos del carrito de compras de la tienda web.

## Opciones de consistencia en Amazon DynamoDB

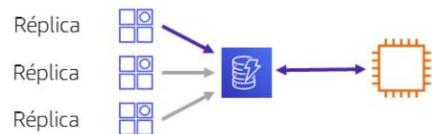


### Consistencia final



La configuración **predeterminada**. En general, todas las copias de los datos alcanzan la consistencia en **un segundo**.

### Consistencia alta



Esta característica es opcional. Utilícela para aplicaciones que requieren que todas las lecturas devuelvan un resultado que refleje todas las escrituras antes de la lectura.

La *consistencia de lectura* representa el *método* y el *tiempo* en el que se refleja un elemento de datos redactado o actualizado correctamente en una operación de lectura posterior para el mismo elemento. Amazon DynamoDB expone la lógica que le permite a usted especificar las características de coherencia que quiere para cada solicitud de lectura en su aplicación.

### Lecturas consistentes finales

Cuando se leen los datos de una tabla de DynamoDB, la respuesta podría no reflejar los resultados de una operación de escritura reciente. La respuesta podría incluir algunos datos anticuados. Si repite la solicitud de lectura tras un breve periodo, la respuesta debería devolver los datos más recientes.

### Lecturas de consistencia alta

Cuando solicita una lectura de consistencia alta, DynamoDB devuelve una respuesta con los datos más actualizados. La respuesta refleja las actualizaciones de todas las operaciones de escritura anteriores que se realizaron correctamente. Una lectura de consistencia alta podría no estar disponible si se produce un retraso o una interrupción en la red.

DynamoDB utiliza las *lecturas consistentes finales* a menos que especifique lo contrario. Las operaciones de lectura (como "GetItem", "Query" y "Scan") proporcionan un parámetro "ConsistentRead". Si establece este parámetro como "true", DynamoDB utiliza lecturas de consistencia alta durante la operación.

Debate de la clase:  
¿Qué base de datos  
debería usar  
la cafetería?

Amazon RDS

Amazon DynamoDB

aws academy

Ahora que conoce las características esenciales clave de Amazon RDS y DynamoDB, considere cuál de estos dos servicios de base de datos sería la mejor opción para el caso de uso de la cafetería.

En un módulo anterior del curso, completó un laboratorio de desafíos donde implementó una base de datos MySQL que se ejecuta en una instancia EC2. Esta base de datos aloja una tabla que contiene detalles sobre el menú de la cafetería. También proporciona la capa de almacenamiento de datos para los pedidos que los clientes realizan en línea.

Si el equipo de la cafetería decide que quiere migrar la capa de base de datos para poder utilizar un servicio de base de datos administrada de Amazon, ¿cuál de estas dos opciones de base de datos sería la mejor opción?

El instructor dirigirá la conversación en la clase sobre este tema. Lo animamos a participar en el diálogo y explicar el razonamiento detrás de su respuesta.

## Aprendizajes clave de la sección 4



The image shows a large, antique-style key lying on a teal-colored wooden surface. Next to the key is a small, white rectangular card with the word "Takeaway" written on it in a black, cursive font.

**aws academy**

- Amazon DynamoDB es un servicio de base de datos NoSQL no relacional completamente administrada **de clave-valor** y documentos.
- DynamoDB **no tiene servidor**, proporciona **escalado horizontal extremo** y **baja latencia**.
- **Las tablas globales** de DynamoDB garantizan que los datos se repliquen en varias regiones.
- DynamoDB proporciona **consistencia final** de forma predeterminada (en general, es totalmente consistente para lecturas un segundo después de la escritura). **La consistencia alta** también es una opción.

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

- Amazon DynamoDB es un servicio de base de datos NoSQL no relacional completamente administrada de clave-valor y documentos.
- DynamoDB no tiene servidor, proporciona escalado horizontal extremo y baja latencia.
- Las tablas globales de DynamoDB garantizan que los datos se repliquen en varias regiones.
- DynamoDB proporciona consistencia final de forma predeterminada (en general, es totalmente consistente para lecturas un segundo después de la escritura). La consistencia alta también es una opción.

**Módulo 5: Incorporación de una capa de base de datos**

**Sección 5: Controles de seguridad  
de la base de datos**



Presentación de la Sección 5: Controles de seguridad de la base de datos.

# Protección de las bases de datos de Amazon RDS



## Recomendaciones

- Ejecute la instancia de RDS en una **nube privada virtual (VPC)**.
  - Proporciona aislamiento de servicios y protección de firewall de IP.
- Utilice las políticas de **AWS Identity and Access Management (IAM)** para la autenticación y el acceso.
  - Los permisos determinan quién puede administrar los recursos de Amazon RDS.
- Utilice **grupos de seguridad** para controlar qué direcciones IP o instancias EC2 pueden conectarse a sus bases de datos.
  - De forma predeterminada, el acceso a la red está desactivado.
- Utilice la **capa de conexión segura (SSL)** para el cifrado en tránsito.
- Utilice el **cifrado** de Amazon RDS en instancias de base de datos e instantáneas para proteger los datos en reposo.
- Utilice las **características de seguridad del motor de base de datos** para controlar quién puede iniciar sesión en las bases de datos de una instancia de base de datos.
- Configure notificaciones de eventos para que se le informe cuando se produzcan eventos importantes de Amazon RDS.



La seguridad es una responsabilidad compartida entre AWS y usted. AWS es responsable de la *seguridad de la nube*, lo que significa que AWS protege la infraestructura que ejecuta Amazon RDS. Mientras que usted es responsable de la *seguridad en la nube*.

Una recomendación de seguridad para Amazon RDS consiste en ejecutar las instancias de RDS en una *nube virtual privada (VPC)*. Una VPC le permite colocar la instancia en una subred privada, que la protege de rutas públicas en Internet. La VPC también ofrece protección de firewall IP y le permite controlar de forma segura la configuración de red aplicable.

Además, le recomendamos que haga lo siguiente:

- Utilice las *políticas de AWS Identity and Access Management (IAM)* para la autenticación y el control del acceso.
- Configure *grupos de seguridad* para limitar las conexiones. Abra el puerto TCP desde donde se puede acceder a la base de datos. Sin embargo, limite desde dónde pueden originarse esas conexiones.
- Utilice *conexiones SSL* para asegurarse de que todas las comunicaciones desde y hacia la base de datos estén protegidas.
- Utilice el *cifrado de Amazon RDS* para cifrar los datos y las instantáneas de la base de datos.
- Utilice las *características de seguridad de su motor de base de datos*, por ejemplo, para imponer la complejidad para contraseñas.

- *Habilite las notificaciones* de eventos importantes que puedan producirse en la instancia de RDS. Estos eventos pueden incluir si la instancia se cerró, si se inició una copia de seguridad, si produjo una conmutación por error, si se cambió el grupo de seguridad o si es poco el espacio de almacenamiento.

## Protección de Amazon DynamoDB



### Recomendaciones

- Utilice los **roles de IAM** para autenticar el acceso
- Utilice las **políticas de IAM**
  - Para definir permisos de acceso detallados para usar API de DynamoDB
  - Defina el acceso al nivel de tablas, elementos o atributos
  - Siga el **principio de concesión de privilegios mínimos**
- Configure **puntos de enlace de la VPC**
  - Evita que el tráfico de conexión atraviese la Internet abierta
  - Las políticas del punto de enlace de la VPC le permiten controlar y limitar el acceso de las API a una tabla de DynamoDB.
- Considere usar **cifrado del lado del cliente**.
  - Cifre datos lo más cerca posible de su origen.



Amazon  
DynamoDB

Seguridad proporcionada de forma predeterminada.

- **Cifrado en reposo** de todos los datos del usuario almacenados en tablas, índices, secuencias y copias de seguridad
- **Cifrado en tránsito**: todas las comunicaciones hacia y desde DynamoDB y otros recursos de AWS utilizan HTTPS

Para proteger Amazon DynamoDB, también se aplican muchas de las mismas prácticas recomendadas que debe utilizar para proteger Amazon RDS.

Por ejemplo, utilice los **roles de IAM** para proteger la autenticación y **utilice las políticas de IAM** para definir permisos de acceso.

Si solo necesita acceder a DynamoDB desde una nube privada virtual (VPC), debe utilizar un **punto de enlace de la VPC** para que solo se pueda acceder desde la VPC requerida. De este modo, se impide que el tráfico atraviese una red de Internet de acceso público y se someta a ese entorno.

Además, si almacena datos confidenciales o delicados en DynamoDB, es buena idea cifrar los datos lo más cerca posible de su origen de modo que sus datos estén protegidos a lo largo de su ciclo de vida.

Tenga en cuenta que DynamoDB proporciona ciertas características de seguridad de forma predeterminada. Por ejemplo, DynamoDB protege los datos del usuario almacenados en reposo y también los datos en tránsito entre clientes en las instalaciones y DynamoDB, y entre DynamoDB y otros recursos de AWS dentro de la misma región de AWS.

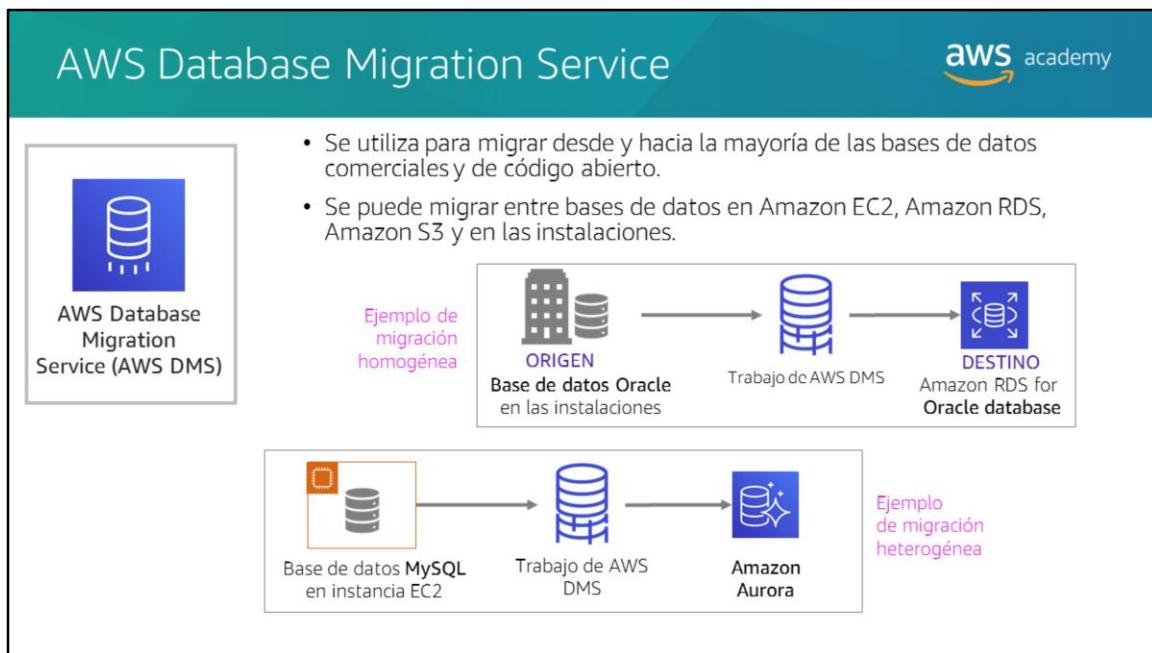
Lea la documentación de [Prácticas recomendadas sobre seguridad de DynamoDB](#) para obtener más detalles.

**Módulo 5: Incorporación de una capa de base de datos**

**Sección 6: Migración de datos a las bases de datos de AWS**



Presentación de la Sección 6: Migración de datos a las bases de datos de AWS.



Puede usar AWS Database Migration Service (AWS DMS) para migrar o replicar bases de datos existentes en Amazon RDS. AWS DMS admite la migración entre las bases de datos más utilizadas.

Las *bases de datos de origen* admitidas incluyen Oracle, Microsoft SQL Server, MySQL, MariaDB, PostgreSQL, IBM Db2 LUW, SAP, MongoDB y Amazon Aurora. Los motores de *base de datos de destino* incluyen Oracle, Microsoft SQL Server, PostgreSQL, MySQL, Amazon Redshift, SAP ASE, Amazon S3 y Amazon DynamoDB.

AWS DMS también admite migraciones homogéneas (mismo motor) y migraciones heterogéneas (motores diferentes).

El primer ejemplo que se muestra es una conversión homogénea, en la que se migra una base de datos Oracle en las instalaciones a Amazon RDS for Oracle Database.

El segundo ejemplo muestra una migración heterogénea, en la que se migra una base de datos MySQL que se ejecuta en una instancia de Amazon EC2 a Amazon Aurora.

## Características clave de AWS DMS



- Realice migraciones por única vez.
- O bien, realice replicación continua de datos.
  - Ejemplo: configure la replicación continua de datos de una base de datos en las instalaciones a una instancia de RDS
- **AWS Schema Conversion Tool (AWS SCT)** admite cambiar el motor de base de datos entre el origen y el destino.
- Pasos principales típicos de la migración:
  1. Crear una base de datos de destino
  2. Migrar el esquema de base de datos
  3. Configurar el proceso de replicación de datos
  4. Iniciar la transferencia de datos y confirmar la finalización
  5. Cambiar la producción a la nueva base de datos (para migraciones únicas)



AWS DMS se puede utilizar para realizar migraciones por única vez, pero también se puede utilizar para realizar replicación continua de datos entre dos bases de datos. Por ejemplo, podría utilizarlo para configurar la replicación continua de datos de una base de datos en las instalaciones a una instancia de RDS.

Cuando desee realizar una migración heterogénea de un motor de base de datos a otro, puede utilizar *AWS Schema Conversion Tool (AWS SCT)*.

Una migración de base de datos típica implica los siguientes pasos principales:

1. Crear una base de datos de destino
2. Migrar el esquema de base de datos
3. Configurar el proceso de replicación de datos
4. Iniciar la transferencia de datos y confirmar la finalización
5. Cambiar la producción a la nueva base de datos (para migraciones únicas)

## Uso de AWS Snowball Edge con AWS DMS



Migrar datos no es práctico en los siguientes casos:

- cuando la base de datos es demasiado grande
- cuando la conexión es demasiado lenta
- cuando hay problemas de privacidad y seguridad

### Utilice **AWS Snowball Edge**

- Transferencia de varios terabytes sin usar Internet



Las migraciones de datos más grandes pueden incluir muchos terabytes de información. Este proceso puede ser difícil debido a los límites de ancho de banda de la red o a la cantidad de datos. AWS Database Migration Service (AWS DMS) puede utilizar [AWS Snowball Edge](#) y Amazon S3 para migrar bases de datos grandes con mayor rapidez que otros métodos.

AWS Snowball es un servicio de AWS que proporciona un dispositivo Edge que puede usar para transferir datos a la nube a velocidades superiores a las de la red. Un dispositivo Edge es un dispositivo propiedad de AWS. Proporciona grandes cantidades de almacenamiento incorporado. El dispositivo Edge usa el cifrado de 256 bits y un módulo de plataforma de confianza (TPM) estándar abierto para garantizar la seguridad y la cadena de custodia completa de los datos.

Cuando se utiliza un dispositivo Edge, el proceso de migración de datos incluye las siguientes etapas:

- Utilice AWS SCT para extraer los datos localmente y transferirlos al dispositivo Edge.
- Envíe el dispositivo de regreso a AWS.
- Despues de que AWS recibe su envío, el dispositivo carga automáticamente sus datos en un bucket de S3.
- AWS DMS toma los archivos y migra los datos al almacén de datos de destino.

## Módulo 5 - Laboratorio de desafíos: Migrar una base de datos a Amazon RDS



Ahora realizará el Módulo 5 - Laboratorio de desafíos: Migrar una base de datos a Amazon RDS.

## Necesidad empresarial: una base de datos administrada



La base de datos que se ejecuta en la instancia EC2 se está volviendo difícil de mantener para Sofía y Nikhil.



Cuando Olivia visitó la cafetería recientemente, les contó sobre las características de Amazon RDS.

Sofía y Nikhil decidieron migrar la base de datos de la cafetería a Amazon RDS.

La administración de la base de datos de la cafetería se está volviendo difícil para Sofía y Nikhil. Por ejemplo, deben trabajar horas extras cuando la cafetería está cerrada para realizar copias de seguridad semanales de la base de datos.

Además, recientemente tuvieron dificultades para instalar un parche requerido. Sofía y Nikhil estuvieron a punto de no terminar durante el período después del cierre del fin de semana. Frank y Martha ahora se dan cuenta de que estas tareas de mantenimiento aumentan los costos de mano de obra del negocio, ya que deben pagar las horas extras que trabajan Sofía y Nikhil.

Raquel, arquitecta de soluciones de AWS, es cliente de la cafetería y amiga de Sofía. Sofía mencionó el tema de las bases de datos a Olivia durante una conversación reciente. Olivia sugirió que migraran la base de datos a Amazon RDS.

Esta solución reduce la carga que supone realizar manualmente tareas de mantenimiento de la base de datos comunes, tales como las copias de seguridad, la instalación de parches y las actualizaciones. Como servicio completamente administrado, Amazon RDS realiza estas tareas de manera automática.

En esta actividad, asumirás los roles de Nikhil y Sofía, y trabajará para migrar la base de datos de la cafetería a Amazon RDS.

## Laboratorio de desafíos: Tareas



1. Creación de una instancia de RDS
2. Análisis de la implementación de las aplicaciones existentes de la cafetería
3. Trabajo con la base de datos en la instancia EC2
4. Trabajo con la base de datos de RDS
5. Importación de los datos a la instancia de la base de datos de RDS
6. Conexión de la aplicación de la cafetería a la nueva base de datos

En este laboratorio de desafíos, realizará las siguientes tareas:

1. Creación de una instancia de RDS
2. Análisis de la implementación de las aplicaciones existentes de la cafetería
3. Trabajo con la base de datos en la instancia EC2
4. Trabajo con la base de datos de RDS
5. Importación de los datos a la instancia de la base de datos de RDS
6. Conexión de la aplicación de la cafetería a la nueva base de datos



El diagrama resume lo que habrá logrado después de completar el laboratorio.

La aplicación web de la cafetería utilizó originalmente el motor MariaDB que está instalado en la instancia EC2. Sin embargo, durante el laboratorio, usted creará una instancia de RDS MariaDB, quitará los datos de la base de datos en la instancia EC2 y migrará los datos a la base de datos de RDS.

Para realizar la migración, se conectará a la instancia EC2 mediante el administrador de sesiones de AWS Systems Manager, que proporciona acceso de terminal a través de un navegador web. Utilizará un cliente MySQL instalado en la instancia EC2 para conectarse a ambas bases de datos, según sea necesario, durante la migración. Las credenciales de la base de datos y la información de conexión de la base de datos se almacenarán en AWS Systems Manager Parameter Store. Por lo tanto, se asociará un rol de IAM a la instancia EC2 para permitir que la aplicación web lea los datos del Almacén de parámetros.



A clock icon with the text "~ 80 minutos" indicates the duration of the module.

The AWS Academy logo is in the top right corner.

**Comenzar con el Módulo 5 - Laboratorio de desafíos: Migrar una base de datos a Amazon RDS**

Ha llegado el momento de comenzar el laboratorio de desafíos.

## Análisis del laboratorio de desafíos: aprendizajes clave



El instructor podría ahora mediar una conversación sobre los aprendizajes clave de este laboratorio de desafíos, una vez que haya terminado.

**Módulo 5: Incorporación de una capa de base de datos**

## Conclusión del módulo



Ha llegado el momento de hacer un repaso del módulo y concluir con una evaluación de conocimientos y un debate sobre una pregunta del examen de certificación como práctica.

## Resumen del módulo



A modo de resumen, en este módulo, aprendió a hacer lo siguiente:

- Comparar tipos de bases de datos
- Diferenciar entre servicios administrados y no administrados
- Explicar cuándo usar Amazon Relational Database Service (Amazon RDS)
- Explicar cuándo usar Amazon DynamoDB
- Describir los controles de seguridad de la base de datos
- Describir cómo migrar datos a bases de datos de Amazon Web Services (AWS)
- Implementar un servidor de base de datos

A modo de resumen, en este módulo, aprendió a hacer lo siguiente:

- Comparar tipos de bases de datos
- Diferenciar entre servicios administrados y no administrados
- Explicar cuándo usar Amazon Relational Database Service (Amazon RDS)
- Explicar cuándo usar Amazon DynamoDB
- Describir los controles de seguridad de la base de datos
- Describir cómo migrar datos a bases de datos de Amazon Web Services (AWS)
- Implementar un servidor de base de datos

## Completar la evaluación de conocimientos



Ha llegado el momento de completar la evaluación de conocimientos de este módulo.

## Pregunta del examen de muestra



Una aplicación requiere una base de datos relacional de alta disponibilidad con una capacidad de almacenamiento inicial de 8 TB. La base de datos crecerá 8 GB todos los días. Para admitir el tráfico esperado, se requerirán al menos ocho réplicas de lectura para manejar las lecturas de la base de datos.

¿Qué opción cumplirá estos requisitos?

- A. DynamoDB
- B. Amazon S3
- C. Amazon Aurora
- D. Amazon Redshift

Observe las opciones de respuesta y descarte las opciones incorrectas en función de las palabras clave que se destacaron antes.

**La respuesta correcta es la “C”.** Amazon Aurora es una base de datos relacional que se escalará automáticamente para adaptarse al crecimiento de los datos. Amazon Redshift no admite réplicas de lectura y no se escalará automáticamente. DynamoDB es un servicio NoSQL, no una base de datos relacional. Amazon S3 es un almacenamiento de objetos, no una base de datos relacional.

## Recursos adicionales



- [Bases de datos de AWS – Página de recursos](#)
- [Guía de introducción de Amazon RDS](#)
- [Prácticas recomendadas de Amazon RDS](#)
- [Preguntas frecuentes sobre Amazon RDS](#)
- [Guía para desarrolladores de Amazon DynamoDB](#)
- [Preguntas frecuentes de Amazon DynamoDB](#)

Si desea obtener más información acerca de los temas que se trataron en este módulo, puede que le resulten útiles los siguientes recursos adicionales:

- [Bases de datos de AWS – Página de recursos](#)
- [Guía de introducción de Amazon RDS](#)
- [Prácticas recomendadas de Amazon RDS](#)
- [Preguntas frecuentes sobre Amazon RDS](#)
- [Guía para desarrolladores de Amazon DynamoDB](#)
- [Preguntas frecuentes de Amazon DynamoDB](#)

# Gracias

© 2020, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. Este contenido no puede reproducirse ni redistribuirse, total ni parcialmente, sin el permiso previo por escrito de Amazon Web Services, Inc. Queda prohibida la copia, el préstamo o la venta de carácter comercial. Envíenos sus correcciones o comentarios relacionados con el curso a: [aws.course.feedback@amazon.com](mailto:aws.course.feedback@amazon.com). Si tiene cualquier otra duda, contáctese con nosotros en: [https://aws.amazon.com/contact-us/aws\\_training/](https://aws.amazon.com/contact-us/aws_training/). Todas las marcas comerciales pertenecen a sus propietarios.



Gracias por completar este módulo.