¿Qué es Amazon Relational  
Database Service (Amazon RDS)?

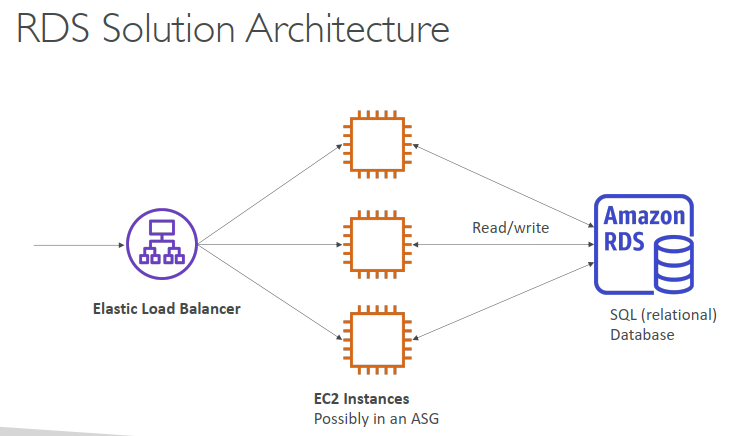
Amazon Relational Database Service (Amazon RDS) es un servicio web que facilita la configuración, la operación y la escala de una base de datos relacional en Nube de AWS. Proporciona una capacidad rentable y de tamaño ajustable para una base de datos relacional estándar y se ocupa de las tareas de administración de bases de datos comunes.

Amazon RDS y Amazon EC2

Amazon RDS es un servicio de base de datos administrada. Es responsable de la mayoría de las tareas de administración. Al eliminar las tediosas tareas manuales, Amazon RDS le permite centrarse en su aplicación y en sus usuarios**. Recomendamos Amazon RDS sobre Amazon EC2 como opción predeterminada para la mayoría de las implementaciones de bases de datos.**

Amazon RDS proporciona las siguientes ventajas específicas sobre las implementaciones de bases de datos que no están completamente administradas:

* Puede utilizar los productos de base de: MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server y Aurora (propietaria de AWS).
* Amazon RDS administra las copias de seguridad, la aplicación de parches de software, la detección automática de errores y la recuperación.
* Puede activar las copias de seguridad automatizadas o crear manualmente sus propias instantáneas de copia de seguridad. Estas copias de seguridad se pueden utilizar para restaurar una base de datos. El proceso de restauración de Amazon RDS es fiable y eficaz.
* **Puede obtener alta disponibilidad con una instancia principal y una instancia secundaria síncrona**, con capacidad de conmutación por error en el caso de que surjan problemas. También puede utilizar réplicas de lectura para aumentar el escalado de lectura.
* Además de la seguridad en el paquete de la base de datos, puede ayudar a controlar quién puede obtener acceso a las bases de datos de RDS mediante AWS Identity and Access Management (IAM) para definir usuarios y permisos. Para ayudar a proteger sus bases de datos, también puede ponerlas en una nube virtual privada (VPC).
* El almacenamiento está respaldado por EBS (gp2 o Io1)

****

Instancias de Base de Datos RDS

Instancias de base de datos

Una *instancia de base de datos*es un entorno de base de datos aislado en la AWS nube. **El componente básico de Amazon RDS es la instancia de base de datos.**

Su instancia de base de datos puede contener una o más bases de datos creadas por el usuario. Puede acceder a su instancia de base de datos utilizando las mismas herramientas y aplicaciones que utiliza con una instancia de base de datos independiente. Puede crear y modificar una instancia de base de datos mediante la **AWS Command Line Interface, la API de Amazon RDS o la AWS Management Console.**

**NOTA:** No se puede acceder mediante SSH a las instancias de bases de datos

Clases de instancia de base de datos

Una *clase de instancia de base de datos* determina la capacidad de cómputo y de memoria de una instancia de base de datos. Una clase de instancia de base de datos consta tanto del tipo de instancia de base de datos como del tamaño. Cada tipo de instancia ofrece diferentes capacidades de computación, memoria y almacenamiento

AWS Regiones y zonas de disponibilidad

Puede ejecutar su instancia de base de datos en varias zonas de disponibilidad, lo que recibe el nombre de despliegue Multi-AZ. Cuando elige esta opción, Amazon aprovisiona automáticamente y mantiene una o más instancias de base de datos secundarias en espera en una zona de disponibilidad diferente. Su instancia de base de datos principal se replica en todas las zonas de disponibilidad para cada instancia de base de datos secundaria. Este enfoque ayuda a proporcionar redundancia de datos y soporte de conmutación por error, elimina los bloqueos de E/S y minimiza los picos de latencia durante las copias de seguridad del sistema. En una implementación de clústeres de base de datos Multi-AZ, las instancias de bases de datos secundarias también pueden ofrecer tráfico de lectura

¿Qué es Amazon Aurora?

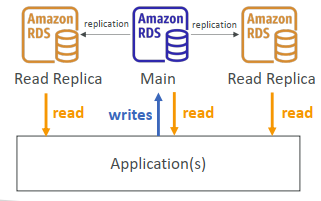
Aurora incluye un subsistema de almacenamiento de alto rendimiento. Sus motores de base de datos compatibles con MySQL y PostgreSQL están personalizados para aprovechar su almacenamiento de rápida distribución. El almacenamiento subyacente crece automáticamente en función de las necesidades. Un volumen de clúster de Aurora puede aumentar hasta un tamaño máximo de 128 tebibytes (TiB). Aurora también automatiza y estandariza la agrupación en clústeres y la reproducción de base de datos, que suelen ser algunos de los aspectos más problemáticos de la configuración y administración de las bases de datos.

Aurora forma parte del servicio de base de datos administrada de Amazon Relational Database Service (Amazon RDS) **y cuesta un 20% más que RDS pero es mas eficiente**

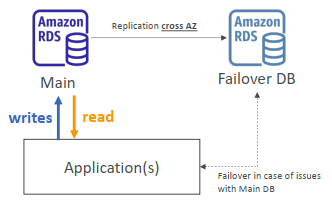
Aurora mejora el rendimiento de MySQL por 5x y el de Postgresql por 3x.

Implementaciones Multi-AZ para alta disponibilidad

Las implementaciones Multi-AZ pueden tener una o dos instancias de base de datos en espera. Cuando la implementación tiene una instancia de base de datos en espera, se denomina implementación de instancia de base de datos Multi-AZ. Una implementación de instancia de base de datos Multi-AZ tiene una instancia de base de datos en espera que proporciona compatibilidad con la conmutación por error, pero no sirve tráfico de lectura. Cuando la implementación tiene dos instancias de base de datos en espera, se denomina implementación de clúster de base de datos Multi-AZ. La implementación de un clúster de base de datos Multi-AZ tiene instancias de base de datos en espera que proporcionan compatibilidad con la conmutación por error y también pueden servir tráfico de lectura.

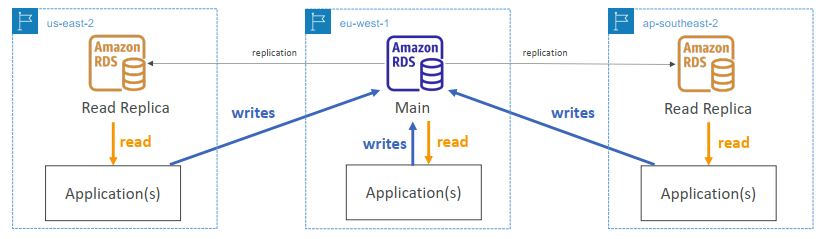
****Réplicas de lectura

* Escalar la carga de lectura de la base de datos
* Se pueden crear hasta 5 bases de datos de lectura
* Los datos son solo escritos en la base de datos principal

Réplicas de Lectura Multi Region

* Recuperación de datos en caso de problemas con una región
* Replicación de costos
* Performance local para lecturas globales

Multi AZ

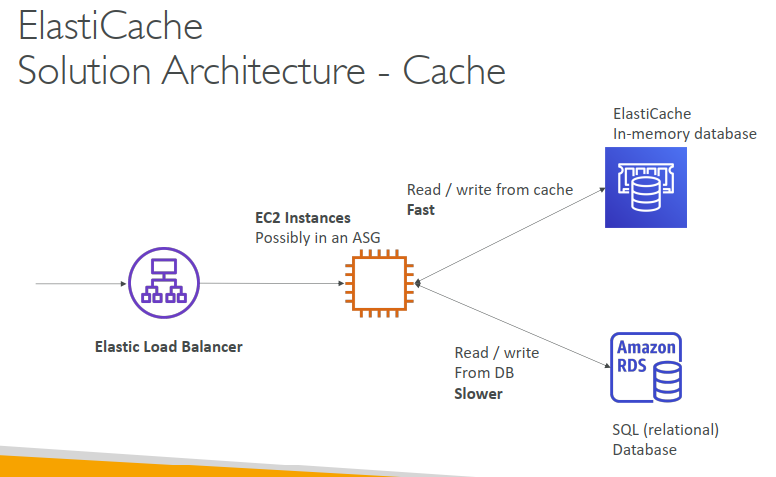
* Se activa en caso de una falla con la AZ donde se aloja mi DB principal
* Solo se puede tener una sola AZ como “failover”
* Los datos son escritos y leidos en la base de datos principal

Amazon ElastiCache Overview

Amazon ElastiCache es un servicio web que facilita la configuración, la administración y el escalado de un almacén de datos en memoria distribuido o de un entorno de caché en la nube: Proporciona una capacidad de almacenamiento en caché de alto rendimiento, escalable y rentable. Al mismo tiempo, ayuda a eliminar la complejidad propia de la implementación y la administración de un entorno de caché distribuido.

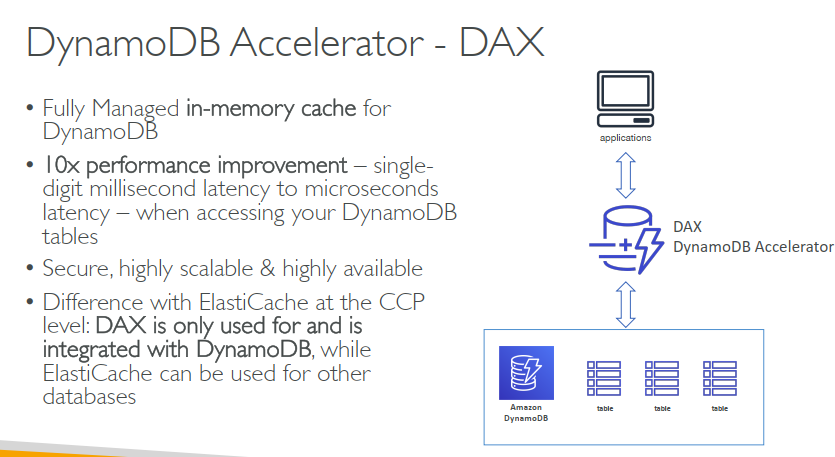
Básicamente, una cache de base de datos, es una BD “en memoria” de alto rendimiento y baja latencia

Ayuda a reducir la carga de la base de datos de operaciones de lectura intensivas, asegurándonos de que las consultas vayan a la cache de ElastiCache



¿Qué es Amazon DynamoDB?

Amazon DynamoDB es un servicio de base de datos NoSQL totalmente administrado que ofrece un rendimiento rápido y predecible, así como una perfecta escalabilidad. DynamoDB le permite delegar las cargas administrativas que supone tener que utilizar y escalar bases de datos distribuidas, para que no tenga que preocuparse del aprovisionamiento, la instalación ni la configuración del hardware, ni tampoco de las tareas de replicación, aplicación de parches de software o escalado de clústeres. DynamoDB también ofrece el cifrado en reposo, que elimina la carga y la complejidad operativa que conlleva la protección de información confidencial

* Base de datos totalmente administrada, altamente disponible y **con replicación a través de 3 zonas de disponibilidad**
* Base de datos No-SQL
* Base de datos distribuida y “serverless”
* **Escala a cargas de trabajo masivas.** Millones de solicitudes por segundo, trillones de filas y cientos de terabytes de almacenamiento
* **Latencia de milisegundos de 1 dígito – low latency retrieval**
* **Integrado con IAM para seguridad, autorización y administración**
* De bajo costo, y con capacidades de auto escalado

Alta disponibilidad y durabilidad

DynamoDB distribuye automáticamente los datos y el tráfico de las tablas entre un número suficiente de servidores para satisfacer sus requisitos de almacenamiento y rendimiento, al mismo tiempo que mantiene un desempeño uniforme y rápido. Todos los datos se almacenan en discos de estado sólido (SSD) y se replican automáticamente en varias zonas de disponibilidad de una región de AWS, con objeto de ofrecer prestaciones integradas de alta disponibilidad y durabilidad de los datos. **Puede utilizar tablas globales para mantener sincronizadas las tablas de DynamoDB en las regiones de AWS.**

# Componentes básicos de Amazon DynamoDB

En DynamoDB se trabaja principalmente con tablas, elementos y atributos. **Una tabla es una colección de elementos y cada elemento es una colección de atributos.** DynamoDB utiliza claves principales para identificar de forma exclusiva cada uno de los elementos de la tabla e índices secundarios para proporcionar mayor flexibilidad a la hora de realizar consultas.

Al crear una tabla, además de asignarle un nombre, debe especificar su clave principal. La clave principal identifica de forma única a cada elemento de la tabla, de manera que no puede haber dos elementos con la misma clave.

DynamoDB admite dos tipos distintos de clave principal:

* **Clave de partición**: una clave principal simple que consta de un solo atributo denominado clave de partición.

DynamoDB utiliza el valor de clave de partición como información de entrada a una función hash interna. El resultado de la función hash determina la partición (almacenamiento físico interno de DynamoDB) donde se almacenará el elemento.

En una tabla que solo tiene una clave de partición, no puede haber dos elementos que tengan el mismo valor de clave de partición.

* **Clave de partición y clave de ordenación**: este tipo de clave se denomina *clave principal compuesta* y consta de dos atributos. El primer atributo es la *clave de partición* y el segundo, la *clave de ordenación*.

DynamoDB utiliza el valor de clave de partición como información de entrada a una función hash interna. El resultado de la función hash determina la partición (almacenamiento físico interno de DynamoDB) donde se almacenará el elemento. Todos los elementos con el mismo valor de clave de partición se almacenan en posiciones contiguas, ordenados según el valor de la clave de ordenación.

En una tabla que tenga una clave de partición y una clave de ordenación, es posible que varios elementos tengan el mismo valor de clave de partición. Sin embargo, esos elementos deben tener valores de clave de ordenación distintos.

Índices secundarios

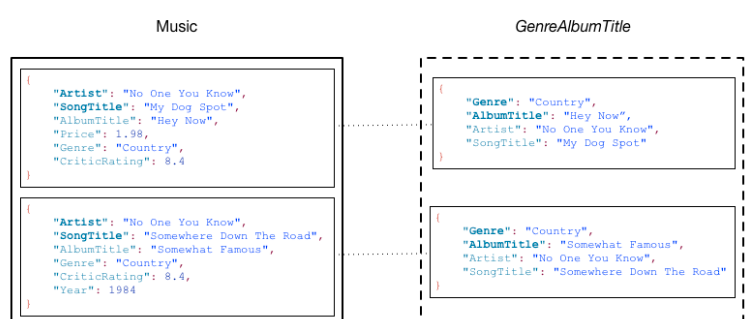
Puede crear uno o varios índices secundarios en una tabla. Un índice secundario le permite consultar los datos de la tabla usando una clave alternativa, además de realizar consultas basadas en la clave principal. DynamoDB no requiere que se usen índices; sin embargo, estos ofrecen a las aplicaciones mayor flexibilidad a la hora de consultar los datos. Después de crear un índice secundario en una tabla, podrá leer los datos en el índice prácticamente de la misma forma que en la tabla.

DynamoDB admite dos tipos de índices:

* Índice secundario global: índice con una clave de partición y una clave de ordenación que pueden diferir de las claves de la tabla.
* Índice secundario local: índice que tiene la misma clave de partición que la tabla, pero una clave de ordenación distinta.

Cada tabla de DynamoDB tiene una cuota de 20 índices secundarios globales (cuota predeterminada) y 5 índices secundarios locales.

En el ejemplo de la tabla Music mostrado anteriormente, puede consultar los elementos de datos por Artist (clave de partición) o por Artist y SongTitle (claves de partición y ordenación). ¿Qué sucede si también desea consultar los datos por género musical (Genre) y título de álbum (AlbumTitle)? Para ello, puede crear un índice basado en Genre y AlbumTitle y, a continuación, consultarlo prácticamente de la misma forma que se consultaría la tabla Music.

****

Consistencia de lectura

Amazon DynamoDB está disponible en varias regiones de AWS de todo el mundo. Cada región es independiente y se encuentra aislada de las demás regiones de AWS. Por ejemplo, si tenemos una tabla denominada People (Personas) en la región us-east-2 y otra tabla denominada People (Personas) en la región us-west-2, se consideran dos tablas completamente independientes.

Cada región de AWS consta de varias ubicaciones distintas denominadas Zonas de disponibilidad. Cada zona de disponibilidad está aislada de los errores que se produzcan en otras zonas de disponibilidad y proporciona conectividad de red de baja latencia económica con otras zonas de disponibilidad de la misma región. Esto permite la replicación rápida de los datos entre varias zonas de disponibilidad de una región.

Cuando la aplicación escribe datos en una tabla de DynamoDB y recibe una respuesta HTTP 200 (OK), la escritura se ha realizado y es duradera. Los datos presentan consistencia final en todas las ubicaciones de almacenamiento, normalmente en el plazo de un segundo o menos.

**DynamoDB admite las lecturas *coherentes finales* y de *consistencia alta*.**

**Lecturas consistentes finales**

Al leer 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 intervalo de tiempo, la respuesta debería devolver los datos más recientes

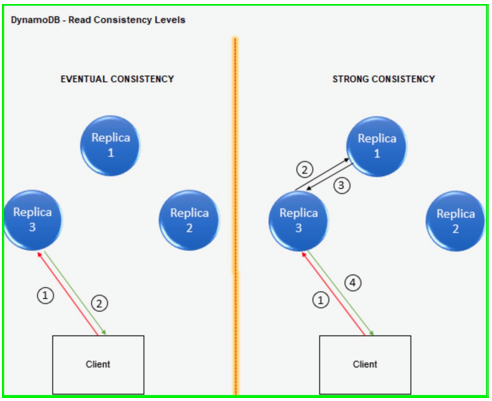
**Lecturas de consistencia alta**

Cuando se solicita una lectura de consistencia alta, DynamoDB devuelve una respuesta con los datos más actualizados, de tal forma que refleja las actualizaciones de todas las operaciones de escritura anteriores que se han llevado a cabo correctamente. No obstante, esta coherencia conlleva algunas desventajas:

* Una lectura de consistencia alta podría no estar disponible si se produce un retraso o una interrupción en la red. En este caso, DynamoDB podría devolver un error de servidor (HTTP 500).
* Las lecturas con coherencia alta pueden tener mayor latencia que las lecturas con coherencia final.
* Las lecturas altamente consistentes no se admiten para los índices secundarios globales.
* Las lecturas con coherencia alta utilizan mayor capacidad de rendimiento que las lecturas con coherencia final.

**Nota**

DynamoDB utiliza las lecturas eventualmente consistentes a menos que especifique lo contrario. Las operaciones de lectura (como GetItem, Queryy Scan) proporcionan un parámetro ConsistentRead. Si establece este parámetro en true (verdadero), DynamoDB utiliza lecturas de consistencia alta durante la operación.

****

Modo de capacidad de lectura/escritura

Amazon DynamoDB posee dos modos de capacidad de lectura/escritura para procesar las lecturas y escrituras en las tablas:

* Bajo demanda
* Capacidad aprovisionada (valor predeterminado, apta para capa gratuita)

El modo de capacidad de lectura/escritura controla cómo se le cobrará el rendimiento de lectura y escritura y cómo se administra la capacidad. Puede establecer el modo de capacidad de lectura/escritura al crear una tabla y también modificarlo más adelante.

Modo bajo demanda

Amazon DynamoDB on-demand es una opción de facturación flexible que permite atender a miles de solicitudes por segundo sin tener que planificar la capacidad. DynamoDB bajo demanda ofrece precios de pago por solicitud para las solicitudes de lectura y escritura. De este modo, únicamente tendrá que pagar por aquello que utilice.

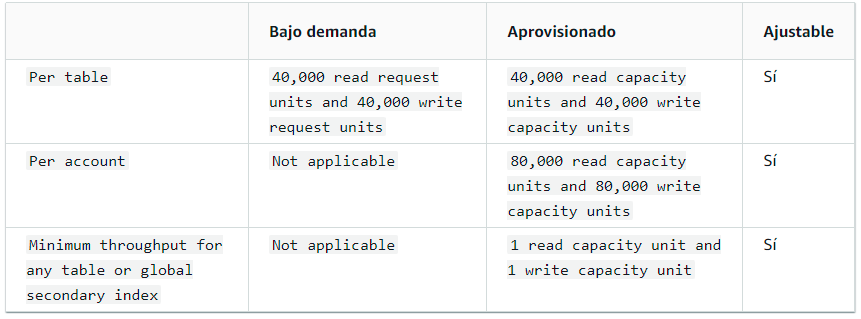
Al elegir el modo on-demand, DynamoDB se adapta de forma instantánea a sus cargas de trabajo a medida que aumentan o disminuyen a cualquier nivel de tráfico alcanzado previamente. Si el nivel de tráfico de una carga de trabajo alcanza un nuevo nivel máximo, DynamoDB se adapta rápidamente para acomodar la carga de trabajo. Las tablas que usan el modo en diferido proporcionan la misma latencia de milisegundos de un solo dígito, compromiso de acuerdo de nivel de servicio (SLA) y seguridad que ya ofrece DynamoDB. Puede elegir el modo en diferido para las tablas nuevas y existentes, y puede seguir usando las API de DynamoDB existentes sin cambiar el código.

El modo bajo demanda es una opción interesante si se da cualquiera de las condiciones siguientes:

* Crea nuevas tablas con cargas de trabajo desconocidas.
* El tráfico de la aplicación es impredecible.
* Prefiere disfrutar de la comodidad de pagar solo por lo que usa.

**La única limitación de la tasa de solicitud son las cuotas predeterminadas de las tablas respecto al rendimiento de DynamoDB, pero se puede elevar bajo petición.**

### Cuotas de rendimiento predeterminadas

****AWS aplica unas cuotas predeterminadas al rendimiento que su cuenta puede aprovisionar y consumir en una región. A continuación, se muestran las cuotas de rendimiento que se aplican en su cuenta, de forma predeterminada.

**NOTA**

Todo el rendimiento disponible para la cuenta se puede aplicar a una sola tabla o a varias tablas.

Unidades de capacidad de lectura y de escritura

Para las tablas en modo aprovisionado, la capacidad de rendimiento se especifica en términos de unidades de capacidad de lectura (RCU) y de escritura (WCU):

Para tablas en modo en diferido, no necesita especificar el rendimiento de lectura y escritura que espera de su aplicación. DynamoDB le carga las lecturas y escrituras que la aplicación lleva a cabo en las tablas en términos de unidades de solicitud de lectura y de escritura, respectivamente.

* Una unidad de solicitud de lectura equivale a una solicitud lectura de consistencia alta, o dos solicitudes de lectura eventualmente consistentes, para un elemento con un tamaño de hasta 4 KB. Dos unidades de solicitud de lectura representan una lectura transaccional respecto a los elementos de hasta 4 KB. Para leer un elemento mayor que 4 KB, DynamoDB necesita unidades de solicitud de lectura adicionales. El número total de unidades de solicitud de lectura necesarias depende del tamaño del elemento y de si se desea utilizar lecturas consistentes finales o de consistencia alta. Por ejemplo, si el tamaño de elemento es de 8 KB, se necesitan 2 unidades de solicitud de lectura para sostener una lectura de consistencia alta, 1 unidad de solicitud de lectura si opta por las lecturas consistentes finales o 4 unidades de solicitud de lectura para una solicitud de lectura transaccional.
* Una unidad de solicitud de escritura equivale a una escritura para un elemento con un tamaño de hasta 1 KB. Para escribir un elemento mayor que 1 KB, DynamoDB tendrá que consumir unidades de solicitud de escritura adicionales. Las solicitudes de escritura transaccionales requieren 2 unidades de solicitud de escritura para realizar una escritura respecto a los elementos de hasta 1 KB. El número total de unidades de solicitud de escritura necesarias depende del tamaño del elemento. Por ejemplo, si el tamaño de elemento es de 2 KB, se necesitan 2 unidades de solicitud de lectura para sostener una solicitud de escritura o 4 unidades de solicitud de lectura para una solicitud de lectura transaccional.

Modo aprovisionado

Si elige el modo aprovisionado, usted especifica el número de lecturas y escrituras por segundo que necesita para su aplicación. Puede utilizar el escalado automático para ajustar automáticamente la capacidad aprovisionada de la tabla en respuesta a los patrones de tráfico. **Esto le ayuda a controlar el uso de DynamoDB para que se mantenga igual o menor que la velocidad de solicitudes definida y, de esta forma, poder predecir los costos.**

El modo aprovisionado es una opción interesante si se da cualquiera de las condiciones siguientes:

* El tráfico de la aplicación es predecible.
* Ejecuta aplicaciones cuyo tráfico es constante o aumenta gradualmente.
* Puede prever los requisitos de capacidad para controlar los costos.

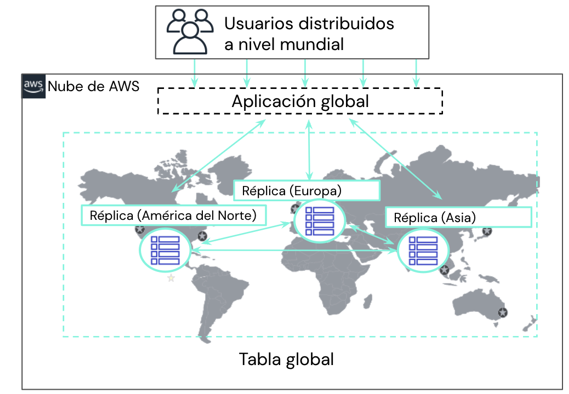
Unidades de capacidad de lectura y de escritura

* Una unidad de capacidad de lectura equivale a una lectura de consistencia alta por segundo, o bien a dos lecturas eventualmente consistentes por segundo, para un elemento con un tamaño de hasta 4 KB. Las solicitudes de lectura transaccionales requieren dos unidades de capacidad de lectura por segundo respecto a los elementos de hasta 4 KB. Para leer un elemento mayor que 4 KB, DynamoDB tiene que consumir unidades de capacidad de lectura adicionales. El número total de unidades de capacidad de lectura necesarias depende del tamaño del elemento y de si se desea utilizar lecturas consistentes finales o de consistencia alta. Por ejemplo, si el tamaño de elemento es de 8 KB, se necesitan 2 unidades de capacidad de lectura para sostener una lectura de consistencia alta por segundo, 1 unidad de capacidad de lectura si opta por las lecturas consistentes finales o 4 unidades de capacidad de lectura para una solicitud de lectura transaccional
* Una unidad de capacidad de escritura equivale a una escritura por segundo para un elemento con un tamaño de hasta 1 KB. Para escribir un elemento mayor que 1 KB, DynamoDB tiene que consumir unidades de capacidad de escritura adicionales. Las solicitudes de escritura transaccionales requieren 2 unidades de capacidad de escritura para realizar una escritura por segundo respecto a los elementos de hasta 1 KB. El número total de unidades de capacidad de escritura necesarias depende del tamaño del elemento. Por ejemplo, si el tamaño de elemento es de 2 KB, se necesitan 2 unidades de capacidad de escritura para sostener una solicitud de escritura por segundo o 4 unidades de capacidad de escritura para una solicitud de escritura transaccional

Tablas globales: replicación en varias regiones para  
DynamoDB

Las tablas globales de Amazon DynamoDB proporcionan una solución completamente administrada para implementar una base de datos en varias regiones y multiactiva sin tener que crear ni mantener su propia solución de replicación. Puede especificar las regiones de AWS en las que desea que las tablas estén disponibles y DynamoDB propagará los cambios de datos en curso a todas ellas.

Las tablas globales de DynamoDB son ideales para aplicaciones con escalabilidad horizontal masiva que cuenten con usuarios de todo el mundo. En esta situación, los usuarios esperan conseguir un rendimiento óptimo de la aplicación. Las tablas globales le proporcionan la opción de replicar varias versiones activas en varias regiones de AWS de todo el mundo. Le permiten ofrecer a los usuarios acceso a los datos de baja latencia, independientemente de dónde se encuentren.



Prácticas recomendadas para el diseño y la arquitectura con DynamoDB

Prácticas recomendadas para diseñar y utilizar claves de partición de forma eficaz

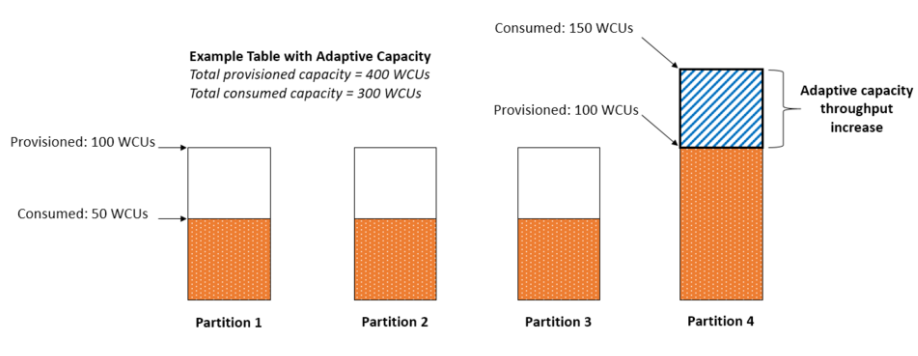
La clave principal que identifica de manera inequívoca cada elemento de una tabla de Amazon DynamoDB puede ser simple (ser únicamente una clave de partición) o compleja (estar formada por una clave de partición y una clave de ordenación).

Por lo general, la aplicación debería diseñarse con la idea de que la actividad será uniforme en todas las claves de las particiones lógicas de la tabla y sus índices secundarios. Puede determinar los patrones de acceso que requiere la aplicación y estimar el total de unidades de capacidad de lectura (RCU) y unidades de capacidad de escritura (WCU) que requiere cada una de las tablas y los índices secundarios.

DynamoDB admite los patrones de acceso utilizando el rendimiento aprovisionado, siempre y cuando el tráfico que llega a una determinada partición no sea superior a 3000 RCU o 1000 WCU.

Aumento de la capacidad de rendimiento en particiones de tráfico  
intenso

No siempre es posible distribuir uniformemente la actividad de lectura y escritura. Si el acceso a los datos está desequilibrado, una partición "caliente" podría recibir un volumen mayor de tráfico de lectura y escritura que otras particiones. En casos extremos, podría llegar a producirse una limitación controlada si una partición recibe más de 3000 RCU o 1000 WCU.



Para acomodar mejor los patrones de acceso desiguales, la capacidad de adaptación de DynamoDB permite que una aplicación pueda seguir leyendo particiones calientes y escribiendo en ellas sin que se produzca una limitación controlada, siempre que el tráfico no supere la capacidad total aprovisionada para la tabla o la capacidad máxima de la partición. De forma automática e instantánea, la capacidad de adaptación hace que la capacidad de rendimiento aumente en las particiones que reciben más tráfico.

Aislamiento de elementos de acceso frecuente

Si su aplicación dirige un tráfico alto hacia uno o dos elementos de forma desproporcionada, la capacidad de adaptación volverá a equilibrar sus particiones de manera que los elementos con acceso frecuente no residan en la misma partición. Este aislamiento de los elementos de acceso frecuente reduce la probabilidad de que se solicite la limitación controlada debida a que la carga de trabajo supere la cuota de rendimiento de una única partición.

Si la aplicación genera habitualmente gran intensidad de tráfico dirigido a un solo elemento, la capacidad de adaptación puede reequilibrar los datos, de tal forma que una partición solo contenga ese elemento al que se accede con frecuencia. En este caso, DynamoDB puede proporcionar rendimiento hasta el máximo de la partición de 3000 RCU o 1000 WCU a la clave principal de ese elemento único. **La capacidad de adaptación no dividirá las colecciones de elementos en varias particiones de la tabla cuando haya un índice secundario local en la tabla.**

**NOTA**

Esta funcionalidad de aislamiento no está disponible para las tablas que usan el modo de capacidad de lectura/escritura aprovisionada que ha habilitado DynamoDB Streams.

