
Migración de bases de datos a Amazon Aurora

Junio de 2016



© 2016, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

Avisos

Este documento se suministra únicamente con fines informativos. Representa la oferta actual de productos y prácticas de AWS a partir de la fecha de publicación de este documento. Dichas prácticas y productos pueden modificarse sin previo aviso. Los clientes son responsables de realizar sus propias evaluaciones independientes de la información contenida en este documento y de cualquier uso de los productos o servicios de AWS, cada uno de los cuales se ofrece “tal cual”, sin garantía de ningún tipo, ya sea explícita o implícita. Este documento no genera ninguna garantía, declaración, compromiso contractual, condición ni certeza por parte de AWS, sus filiales, proveedores o licenciantes. Las responsabilidades y obligaciones de AWS con respecto a sus clientes se controlan mediante los acuerdos de AWS y este documento no forma parte ni modifica ningún acuerdo entre AWS y sus clientes.

Contenido

Resumen	4
Introducción a Amazon Aurora	4
Consideraciones de cara a la migración de datos	6
Fases de la migración	6
Consideraciones sobre las aplicaciones	7
Consideraciones sobre particiones horizontales y réplica de lectura	8
Consideraciones sobre fiabilidad	9
Consideraciones sobre licencias y costos	9
Otras consideraciones de cara a la migración	10
Planificación de los procesos de migración de las bases de datos	11
Migración homogénea	11
Migración heterogénea	13
Migración de bases de datos de gran tamaño a Amazon Aurora	14
Consolidación de particiones funcionales y horizontales en Amazon Aurora	15
Panorámica de las opciones de cara a la migración de datos	17
Migración de instantánea de RDS	17
Migración del esquema de la base de datos	23
Migración de esquema homogénea	24
Migración de esquema heterogénea	25
Migración de esquema con la herramienta de conversión de esquemas de AWS	26
Migración de datos	36
Introducción y enfoque general para AWS DMS	36
Métodos de migración	37
Procedimiento de migración	38
Pruebas y puesta en servicio	44

Pruebas de migración	44
Puesta en servicio	45
Conclusión	46
Colaboradores	47
Documentación adicional	47
Notas	47

Resumen

Amazon Aurora es un motor de bases de datos relacionales compatible con MySQL pensado para empresas. Amazon Aurora es una base de datos diseñada para la nube que deja atrás muchas de las limitaciones de los tradicionales motores de bases de datos relacionales. Este documento técnico pretende señalar las prácticas recomendadas a la hora de migrar sus bases de datos a Amazon Aurora. El documento presenta consideraciones de cara a la migración a Amazon Aurora y el proceso paso a paso para bases de datos, tanto de código abierto como comerciales interfiriendo lo menos posible en el funcionamiento de las aplicaciones.

Introducción a Amazon Aurora

Durante décadas, las bases de datos relacionales convencionales han sido la alternativa de primera elección a la hora de usar soluciones de persistencia y almacenamiento de datos. Estos sistemas de bases de datos siguen dependiendo de arquitecturas monolíticas, y no se diseñaron para aprovechar la infraestructura de la nube. Estas arquitecturas monolíticas presentan muchos retos, especialmente en lo relativo al costo, a la flexibilidad y a la disponibilidad. Para abordar estos desafíos, AWS ha rediseñado la base de datos relacional para adaptarla a la infraestructura de la nube, presentando Amazon Aurora.

Amazon Aurora es un motor de bases de datos relacionales compatible con MySQL que combina la velocidad, seguridad y disponibilidad de las bases de datos comerciales de gama alta con la sencillez y la rentabilidad de las bases de datos de código abierto. Aurora ofrece un rendimiento hasta cinco veces mayor que MySQL y un rendimiento comparable al de las bases de datos

comerciales de gama alta. Amazon Aurora se comercializa a un precio diez veces menor que el de los motores comerciales.

Amazon Aurora está disponible a través de la plataforma de Amazon Relational Database Service (Amazon RDS). Al igual que otras bases de datos de Amazon RDS, Aurora es una base de datos totalmente administrada. Con la plataforma de Amazon RDS, la mayoría de las tareas de administración de bases de datos, como por ejemplo el aprovisionamiento de hardware, la aplicación de parches de software, la instalación y configuración, la monitorización y la realización de copias de seguridad están completamente automatizadas.

Amazon Aurora está diseñado para soportar cargas de trabajo de vital importancia y ofrece una disponibilidad muy elevada de forma predeterminada. Un clúster de base de datos de Aurora abarca varias zonas de disponibilidad de una región, proporcionando desde el primer momento durabilidad y tolerancia a fallos para sus datos en los distintos centros de datos físicos. Una zona de disponibilidad se compone de uno o varios centros de datos de alta disponibilidad gestionados por Amazon. Las zonas de disponibilidad están aisladas entre sí, y permanecen conectadas por medio de enlaces de baja latencia. Cada segmento del volumen de su base de datos se replica seis veces por todas estas zonas de disponibilidad.

Los volúmenes de clúster de Aurora aumentan automáticamente a medida que crece la cantidad de datos en su base de datos, sin que ello afecte al desempeño ni a la disponibilidad, por lo que no es necesario realizar estimaciones ni provisiones de almacenamiento para grandes cantidades de datos con antelación. Un volumen de clúster de Aurora puede crecer hasta un tamaño máximo de 64 terabytes (TB). Se le cobrará únicamente por el espacio que utilice en un volumen de clúster de Aurora.

La función de copia de seguridad automatizada de Aurora admite la recuperación de datos a un momento dado en el tiempo, lo que le permite restaurar la base de datos a su estado en cualquier segundo dentro del período de retención, hasta los últimos cinco minutos. Las copias de seguridad automáticas se almacenan en Amazon Simple Storage Service (Amazon S3), que está diseñado para ofrecer una durabilidad del 99,999999999%. Las copias de seguridad de Amazon Aurora son automáticas, incrementales y continuas, y no inciden en el rendimiento de la base de datos.

Para las aplicaciones que necesitan réplicas de solo lectura, se puede crear un máximo de 15 réplicas de Aurora por cada base de datos de Aurora con un retraso de réplica muy corto. Estas réplicas comparten el mismo almacenamiento subyacente que la instancia de origen, lo que reduce los costos y elimina la necesidad de llevar a cabo escrituras en los nodos de cada réplica.

Amazon Aurora es muy seguro, y le permite cifrar sus bases de datos mediante claves que se pueden crear y controlar a través del AWS Key Management Service (AWS KMS). En una instancia de base de datos que se ejecute con cifrado de Amazon Aurora, los datos en reposo que se encuentran en el almacenamiento subyacente están cifrados, al igual que las copias de seguridad, réplicas e instantáneas automatizadas del mismo clúster. Amazon Aurora utiliza SSL (AES-256) para proteger los datos en tránsito.

Para obtener una lista completa de las características de Aurora, consulte la página del producto [Amazon Aurora](#)¹. Dado el amplio conjunto de características y la rentabilidad de Amazon Aurora, se considera cada vez más como la base de datos de referencia para aplicaciones de vital importancia.

Consideraciones de cara a la migración de datos

Una base de datos representa un componente crítico en la arquitectura de la mayoría de las aplicaciones. La migración de la base de datos a una nueva plataforma es un acontecimiento importante en el ciclo de vida de una aplicación, y pueden afectar a la funcionalidad de la misma, así como a su rendimiento y fiabilidad. Debe considerar algunos aspectos importantes antes de iniciar su primer proyecto de migración a Amazon Aurora.

Fases de la migración

Dado que las migraciones de bases de datos suelen ser complejas, recomendamos abordarlas de forma iterativa y gradual.

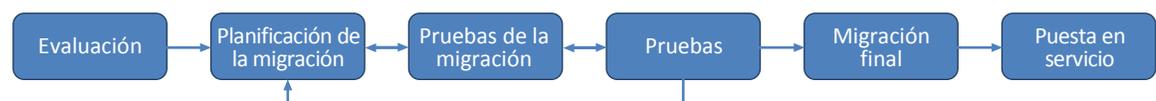


Figura 1: Fases de la migración

Consideraciones sobre las aplicaciones

Evaluación de las características de Aurora

Aunque la mayoría de las aplicaciones se pueden proyectar para trabajar con varios motores de bases de datos relacionales, debe asegurarse de que la aplicación funciona con Amazon Aurora. Amazon Aurora está diseñado para ser totalmente compatible con MySQL 5.6. Por lo tanto, la mayor parte del código, de las aplicaciones, de los controladores y de las herramientas que se utilizan en la actualidad con bases de datos MySQL también se puede utilizar con Aurora realizando pocos cambios o ninguno.

Sin embargo, determinadas características de MySQL, como el motor de almacenamiento MyISAM, no están disponibles con Amazon Aurora. Además, debido a que Aurora es un servicio administrado por naturaleza, el acceso SSH a los nodos de la base de datos está restringido, lo que puede afectar a su capacidad para instalar herramientas o complementos de terceros en el host de la base de datos.

Consideraciones de rendimiento

El rendimiento de la base de datos es un factor clave a la hora de migrar una base de datos a una nueva plataforma. Por lo tanto, muchos proyectos de migración de base de datos que se concluyen con éxito empiezan con un análisis del rendimiento de la nueva plataforma de la base de datos. Aunque las [comparativas de desempeño Sysbench y TPC-C](#) nos pueden dar una idea bastante fiable del rendimiento general de la base de datos, estas pruebas no emulan los patrones de acceso de sus propias aplicaciones. Para obtener resultados más útiles, debe probar el rendimiento de la base de datos para cargas de trabajo sujetas a limitaciones temporales realizando sus consultas (o un subconjunto de estas) directamente en la nueva plataforma.

Considere estas estrategias:

- Si la base de datos es MySQL, debe migrar a Amazon Aurora programando un tiempo de inactividad y probando el rendimiento de la base de datos con una versión de prueba o secuenciada de la aplicación, o bien reproduciendo la carga de trabajo normal de producción.
- Si actualmente funciona con un motor compatible con MySQL, puede copiar de manera selectiva las tablas con una mayor cantidad de datos

a Amazon Aurora y probar en ellas sus consultas. Esto le ofrece un buen punto de partida. Desde luego, las pruebas realizadas después de completar la migración de datos proporcionarán una panorámica completa del rendimiento real de la aplicación en la nueva plataforma.

Amazon Aurora ofrece un rendimiento similar al de otros motores comerciales y notablemente superior al de MySQL, lo que se consigue gracias a una estrecha integración del motor de base de datos con una capa virtualizada de almacenamiento basada en SSD y diseñada para cargas de trabajo de bases de datos. Esto reduce el número de escrituras en el sistema de almacenamiento, minimiza la contención de bloqueos y elimina los retrasos que ocasionan los subprocesos de la base de datos. Nuestras pruebas con SysBench en instancias r3.8xlarge revelan que Amazon Aurora ofrece 585 000 lecturas y 107 000 escrituras por segundo, lo que supone cinco veces más que MySQL cuando ejecuta la misma comparativa de desempeño en el mismo hardware.

Un aspecto en el que Amazon Aurora muestra mejoras significativas respecto a MySQL son las cargas de trabajo con numerosas operaciones simultáneas. Para maximizar el rendimiento de la carga de trabajo en Amazon Aurora, le aconsejamos proyectar sus aplicaciones de manera que manejen una gran cantidad de consultas simultáneas.

Consideraciones sobre particiones horizontales y réplica de lectura

Si la base de datos actual está estructurada en particiones horizontales entre varios nodos, quizás pueda combinar estas particiones horizontales en una única base de datos de Aurora durante la migración. Una sola instancia de Amazon Aurora puede ampliarse hasta 64 TB, admite miles de tablas y un número significativamente superior de lecturas y escrituras respecto a una base de datos MySQL estándar.

Si su aplicación hace un uso intensivo de operaciones de lectura/escritura, considere la posibilidad de utilizar las réplicas de lectura de Aurora para rebajar la carga de operaciones de solo lectura del nodo principal de la base de datos. Esto puede mejorar la concurrencia de la base de datos principal para escrituras, así como el rendimiento global de lectura y escritura. Asimismo, el uso de réplicas de lectura puede reducir sus costos en una configuración de zonas de disponibilidad múltiples (Multi-AZ), ya que es posible utilizar instancias más pequeñas para su instancia principal al tiempo que añade la capacidad

de conmutación por error en el clúster de la base de datos. Las réplicas de lectura de Aurora proporcionan un retraso de réplica prácticamente igual a cero, y puede crear hasta 15 réplicas de lectura.

Consideraciones sobre fiabilidad

Una disponibilidad elevada y la capacidad de recuperación de desastres constituyen aspectos importantes que se deben considerar en las bases de datos. Determine los requisitos para el objetivo de tiempo de recuperación (RTO) y el objetivo de punto de recuperación (RPO) de su aplicación. Con Amazon Aurora puede mejorar significativamente ambos factores.

Amazon Aurora reduce el tiempo de reinicio de la base de datos a menos de 60 segundos en la mayoría de las situaciones de caída de la base de datos. Además, Aurora saca la caché del búfer del proceso de la base de datos, y la habilita inmediatamente al completar el reinicio. En el caso poco probable de sufrir un fallo simultáneo del hardware y de la zona de disponibilidad, la plataforma de la base de datos gestiona automáticamente el proceso de recuperación.

Aurora está diseñado para ofrecerle una recuperación con un RPO igual a cero dentro de una región de AWS, lo que constituye una importante mejora respecto a los sistemas de bases de datos in situ. Aurora mantiene seis copias de los datos en tres zonas de disponibilidad, y trata de recuperar automáticamente la base de datos en una zona de disponibilidad en buen estado sin que se pierdan datos. En el caso improbable de que los datos no se encuentren disponibles en el almacenamiento de Amazon Aurora, puede restablecerlos a partir de una instantánea de la base de datos o realizar una operación de restauración a un determinado punto en el tiempo en una instancia nueva.

Consideraciones sobre licencias y costos

La gestión de una base de datos de su propiedad conlleva costos asociados. Antes de planificar la migración de una base de datos, resulta esencial llevar a cabo un análisis del costo total de propiedad (TCO) de la nueva plataforma. En condiciones ideales, la migración a una nueva plataforma de la base de datos debería reducir el costo total de propiedad a la vez que proporciona características similares o incluso mejores a sus aplicaciones. Si utiliza un motor de base de datos de código abierto (MySQL, Postgres), los costos están relacionados en buena parte con el hardware y las actividades de gestión de los

servidores y de la base de datos. Sin embargo, si está ejecutando un motor de base de datos comercial (Oracle, SQL Server, DB2, etc.), una parte importante del costo deriva de las licencias de la base de datos.

Como Aurora está disponible a una décima parte del costo de los motores comerciales, muchas de las aplicaciones que pasan a Aurora pueden reducir de forma significativa su costo total de propiedad. Incluso si su base de datos se ejecuta en un motor de código abierto, como MySQL o Postgres, puede obtener ahorros significativos gracias a la instalación de Amazon Aurora gracias a su elevado rendimiento y a sus réplicas de lectura, que cumplen un doble cometido. Para obtener más información, visite la página [Precios de Amazon RDS para Aurora](#)².

Otras consideraciones de cara a la migración

Cuando haya terminado de valorar la idoneidad, el rendimiento y la fiabilidad de la aplicación, debe pensar en todo lo que le hará falta para migrar a la nueva plataforma.

Estimación del esfuerzo necesario para cambiar el código

Es importante estimar la cantidad de cambios que necesitará llevar a cabo en el código y en el esquema cuando realice la migración de la base de datos a Amazon Aurora. Cuando la migración se realiza desde bases de datos compatibles con MySQL, la cantidad de cambios en el código que se necesitan es inapreciable. Sin embargo, cuando la migración se realiza desde motores distintos a MySQL, es posible que necesite efectuar cambios en el esquema y en el código. La herramienta de conversión de esquemas de AWS, [de la que nos ocupamos más adelante en este documento](#), puede ayudar a realizar a calcular dicho esfuerzo.

Disponibilidad de las aplicaciones durante la migración

A la hora de realizar una migración a Amazon Aurora, puede adoptar dos enfoques distintos: reservar un tiempo de inactividad predecible para su aplicación o reducir al mínimo indispensable dicho tiempo. El enfoque que elija depende del tamaño de la base de datos y de los requisitos de disponibilidad de sus aplicaciones. En cualquier caso, es una buena idea considerar el impacto del proceso de migración en la aplicación y en su negocio antes de comenzar una migración de la base de datos. Ambos enfoques se detallan en las próximas secciones.

Planificación de los procesos de migración de las bases de datos

En la sección anterior se han tratado algunas de las consideraciones que hay que tener en cuenta a la hora de realizar una migración de bases de datos a Amazon Aurora. Cuando haya determinado que Aurora es la solución adecuada para su aplicación, el siguiente paso consiste en adoptar una estrategia preliminar de migración y crear un plan para llevarla a cabo.

Migración homogénea

Si la base de datos de origen es compatible con MySQL 5.6 (MySQL, MariaDB, Percona, etc.), la migración a Aurora es muy sencilla.

Migración homogénea con tiempo de inactividad

Si la aplicación puede tolerar un tiempo de inactividad durante las horas de baja actividad, una migración de este tipo constituye la opción más sencilla y un enfoque muy recomendable. La mayoría de los proyectos de migración de bases de datos se encuentran en esta categoría, ya que la mayoría de las aplicaciones ya prevén un margen de tiempo para mantenimiento bien definido. Dispone de las siguientes opciones para migrar la base de datos con tiempo de inactividad.

- **Migración de instantánea de RDS** – Si la base de datos de origen se ejecuta en MySQL 5.6 de Amazon RDS, no tiene más que migrar una instantánea de la base de datos a Amazon Aurora. Para las migraciones con tiempos de inactividad, tendrá que detener la aplicación, o bien dejar de escribir en la base de datos mientras crea la instantánea y lleva a cabo el proceso de migración. El tiempo necesario para realizar la migración depende sobre todo del tamaño de la base de datos, y se puede determinar con antelación realizando una migración de prueba. La opción para realizar una instantánea de migración se explica en la sección [Migración de instantáneas de RDS](#). También puede realizar una migración con un tiempo de inactividad prácticamente nulo por medio de la migración de instantáneas y la replicación de logs binarios, proceso que se describe en la siguiente sección.
- **Migración a través de herramientas nativas de MySQL**. Puede utilizar herramientas nativas de MySQL para migrar a Aurora sus datos y su esquema. Se trata de una opción excelente si necesita un mayor control

sobre el proceso de migración de la base de datos, si se encuentra más cómodo manejando herramientas nativas de MySQL o bien si otros métodos de migración no funcionan tan bien en su caso particular. Para las prácticas recomendadas al utilizar esta opción, descargue el documento técnico [Amazon RDS for Aurora Export/Import Performance Best Practices](#)³.

- **Migración mediante AWS Database Migration Service (AWS DMS).** La migración en una sola fase mediante AWS DMS constituye otra herramienta para trasladar su base de datos de origen a Amazon Aurora. Antes de utilizar AWS DMS para trasladar los datos, debe copiar el esquema de la base de datos de origen a la de destino mediante herramientas diseñadas para MySQL. Para conocer el proceso paso a paso, consulte la sección [Migración de datos](#). El uso de AWS DMS representa una opción excelente cuando no posee experiencia en el uso de herramientas nativas de MySQL.

Migración homogénea con un tiempo de inactividad prácticamente nulo

En algunos casos, es posible que desee migrar su base de datos a Aurora con el mínimo tiempo de inactividad posible. A continuación se ofrecen dos ejemplos:

- Cuando la base de datos es relativamente grande y la migración que se sirve de las opciones de inactividad lleva más tiempo del que prevé su margen de tiempo para el mantenimiento de aplicaciones
- Cuando desea ejecutar las bases de datos de origen y de destino en paralelo con fines de pruebas

En estos casos, puede replicar los cambios de la instancia de la base de datos de origen MySQL a Aurora en tiempo real a través de la replicación. Dispone de un par de opciones a elegir entre:

- **Migración con un tiempo de inactividad prácticamente nulo mediante la replicación de logs binarios de MySQL.** Amazon Aurora es compatible con la replicación clásica de logs binarios de MySQL. Si está utilizando una base de datos MySQL, lo más probable es que ya esté familiarizado con la configuración de la replicación clásica de logs binarios. De ser así, y si desea un mayor control del proceso de migración, la carga de la base de datos en una sola fase utilizando las herramientas originales,

junto con la replicación de logs binarios, le ofrece un proceso de migración a Aurora que le resultará conocido.

- **Migración con un tiempo de inactividad prácticamente nulo mediante AWS Database Migration Service (AWS DMS).** AWS DMS no solo es capaz de gestionar la migración en una sola fase, sino que también admite la replicación de datos en tiempo real a través de la captura de cambios de datos (CDC) entre la base de datos de origen y la de destino. AWS DMS se ocupa de los aspectos más complejos relacionados con la copia de datos inicial, la configuración de las instancias de replicación y la monitorización del proceso de replicación. Una vez se haya completado la migración inicial de la base de datos, la base de datos de destino permanecerá sincronizada con la de origen tanto tiempo como desee. Si no está familiarizado con la replicación de logs binarios, AWS DMS es la mejor opción para llevar a cabo migraciones a Amazon Aurora homogéneas y con un tiempo de inactividad prácticamente nulo. Consulte la sección [Introducción y enfoque general a AWS DMS](#).
- **Migración con un tiempo de inactividad prácticamente nulo mediante instantánea de RDS y replicación de logs binarios de MySQL.** Si la base de datos de origen se ejecuta en MySQL 5.6 de Amazon RDS, puede migrar una instantánea a Amazon Aurora y comenzar la replicación de logs binarios entre las bases de datos de origen y destino después de la migración de la instantánea. Para obtener más información sobre esta opción de migración, consulte [Replication with Amazon Aurora](#) en el documento *Amazon RDS User Guide*⁴.

Migración heterogénea

Si lo que busca es migrar una base de datos compatible con los motores de bases de datos MySQL (Oracle, SQL Server PostgreSQL etc.) a Amazon Aurora, puede servirse de varias opciones para llevar a cabo esta de migración de forma rápida y sencilla.

Migración de esquema

La migración de esquema desde una base de datos compatible con MySQL a Amazon Aurora se puede conseguir mediante la herramienta de conversión de esquemas de AWS. Esta herramienta es una aplicación de escritorio que le ayuda a convertir su esquema desde una base de datos Oracle, Microsoft SQL Server o PostgreSQL a una instancia de MySQL de Amazon RDS o a un clúster de base de

datos de Amazon Aurora. En los casos en que el esquema de la base de datos de origen no se pueda convertir en su totalidad de forma automática, la herramienta de conversión de esquemas de AWS le guiará a la hora de crear el esquema de destino equivalente en la base de datos de Amazon RDS. Para obtener más información, consulte la sección [Migración del esquema de la base de datos](#).

Migración de datos

Además de ser compatible con migraciones homogéneas con un tiempo de inactividad prácticamente nulo, AWS Database Migration Service (AWS DMS) también admite la replicación continua entre bases de datos heterogéneas, lo que constituye una de las opciones preferidas a la hora trasladar la base de datos de origen a la de destino, tanto para migraciones con un tiempo de inactividad programado como para aquellas con un tiempo de inactividad prácticamente nulo. Cuando comienza la migración, AWS DMS administra todos los aspectos más complejos del proceso, como la transformación del tipo de datos, la compresión y la transferencia paralela (para realizar una transferencia de datos más rápida), a la vez que garantiza que los cambios que se produzcan durante el proceso de migración en la base de datos de origen se repliquen automáticamente en la de destino.

Además de AWS DMS, puede utilizar varias herramientas de terceros como Attunity Replicate, Tungsten Replicator u Oracle Golden Gate, entre otras, para migrar sus datos a Amazon Aurora. Sea cual sea la herramienta que elija, considere el rendimiento y los costos de licencia antes de completar el conjunto de herramientas que usará para la migración.

Migración de bases de datos de gran tamaño a Amazon Aurora

La migración de conjuntos de datos de gran tamaño presenta retos únicos en cada proyecto. Muchos proyectos de migración de bases de datos de gran tamaño que consiguen finalizarse con éxito utilizan una combinación de las siguientes estrategias:

- **Migración con replicación continua.** Por lo general, las bases de datos de gran tamaño establecen tiempos de inactividad más largos para mover los datos de origen a destino. Para reducir el tiempo de inactividad, puede cargar en primer lugar los datos de referencia en la base de datos de destino y, a continuación, activar la replicación (utilizando herramientas

diseñadas para MySQL o bien AWS DMS o herramientas de terceros) para actualizar los cambios.

- **Copia de las tablas estáticas en primer lugar.** Si la base de datos se apoya en tablas estáticas con datos de referencia de gran tamaño, puede migrar estas tablas a la base de datos de destino antes de efectuar la migración del conjunto de datos activo. Puede utilizar AWS DMS para copiar tablas de manera selectiva o bien exportar e importarlas a mano.
- **Migración en varias fases.** La migración de bases de datos de gran tamaño con miles de tablas se puede dividir en varias fases. Por ejemplo, puede mover un conjunto de tablas sin consultas de combinación cruzada cada fin de semana hasta que la base de datos de origen se haya migrado por completo a la de destino. Tenga en cuenta que, para lograrlo, tendrá que realizar cambios en su aplicación para conectarse a dos bases de datos de forma simultánea al tiempo que su conjunto de datos se encuentra en dos nodos distintos. Aunque no se trata de un patrón de migración común, es una opción más.
- **Limpieza de la base de datos.** Muchas bases de datos de gran tamaño contienen datos y tablas que no se utilizan. En muchos casos, los desarrolladores y administradores mantienen copias de seguridad de las tablas en la propia base de datos, o simplemente se olvidan de soltar las tablas que no se utilizan. Sea cual sea el motivo, un proyecto de migración de base de datos le ofrece una oportunidad para limpiar la base de datos existente antes de la migración. Si algunas tablas no se utilizan, puede soltarlas o archivarlas en otra base de datos. También puede eliminar datos obsoletos de tablas de gran tamaño o bien archivar los datos en ficheros.

Consolidación de particiones funcionales y horizontales en Amazon Aurora

Si ejecuta varias particiones de tipo horizontal o bien funcional de su base de datos para lograr un alto rendimiento, tiene la oportunidad de consolidarlas en una única base de datos de Aurora. Una sola instancia de Amazon Aurora puede ampliarse hasta 64 TB, admite miles de tablas y un número significativamente superior de lecturas y escrituras respecto a una base de datos MySQL estándar. Consolidar estas particiones en una única instancia de Aurora no solo reduce el costo total de propiedad y simplifica la gestión de la base de datos, sino que

también mejora de forma significativa el rendimiento de las consultas entre distintas particiones.

- **Particiones funcionales.** El particionamiento funcional consiste en dedicar diferentes nodos a tareas distintas. Por ejemplo, en una aplicación de comercio electrónico, puede que tenga un nodo de las bases de datos dedicado a los datos del catálogo de productos, y otro nodo para el registro y el procesamiento de pedidos. Como resultado, estas particiones suelen tener esquemas distintos y no superpuestos.

Estrategia de consolidación. Migre cada una de las particiones funcionales como un esquema único a la instancia de destino de Aurora. Si la base de datos de origen es compatible con MySQL, utilice herramientas diseñadas para MySQL para migrar el esquema y, a continuación, utilice AWS DMS para migrar los datos, ya sea de una sola vez o de forma continua mediante la replicación. Si la base de datos de origen no es compatible con MySQL, utilice la herramienta de conversión de esquemas de AWS para migrar los esquemas a Aurora y AWS DMS para realizar una carga única o la replicación continua.

- **Particiones horizontales de datos.** Si utiliza el mismo esquema con diferentes conjuntos de datos repartidos entre varios nodos, está aprovechando el particionamiento horizontal de la base de datos. Por ejemplo, un servicio de gestión de blogs con un tráfico elevado puede crear varias particiones horizontales para la actividad de los usuarios y los datos y, al mismo tiempo, mantener el mismo esquema de tablas.

Estrategia de consolidación. Dado que todas las particiones horizontales comparten el mismo esquema de base de datos, tendrá que crear el esquema de destino tan solo una vez. Si utiliza una base de datos compatible con MySQL, utilice herramientas diseñadas para este sistema para migrar el esquema de la base de datos a Aurora. Si utiliza una base de datos distinta a MySQL, utilice la herramienta de conversión de esquemas de AWS para migrar el esquema de la base de datos a Aurora. Cuando el esquema de la base de datos se haya migrado, es mejor detener las escrituras en las particiones horizontales de la base de datos y utilizar herramientas diseñadas para la base de datos de origen, o bien realizar una carga única de datos con AWS DMS para migrar una determinada partición horizontal de Aurora. Si no puede interrumpir las escrituras en la aplicación durante un largo periodo, puede seguir utilizando AWS DMS con la replicación, pero solo después de haber realizado una planificación y unas pruebas adecuadas.

Panorámica de las opciones de cara a la migración de datos

Tipo de base de datos de origen	Migración con tiempo de inactividad	Migración con tiempo de inactividad prácticamente nulo
MySQL de Amazon RDS	<p>Opción 1: Migración de instantánea de RDS</p> <p>Opción 2: Migración manual con herramientas diseñadas para la base de datos de origen*</p> <p>Opción 3: Migración de esquema con herramientas diseñadas para la base de datos de origen y carga de datos mediante AWS DMS</p>	<p>Opción 1: Migración con herramientas diseñadas para la base de datos de origen + replicación de logs binarios</p> <p>Opción 2: Migración de instantánea de RDS + replicación de logs binarios</p> <p>Opción 3: Migración de esquema con herramientas diseñadas para la base de datos de origen + AWS DMS para transferir los datos</p>
MySQL Amazon EC2 o en las propias instalaciones	<p>Opción 1: Migración con herramientas diseñadas para la base de datos de origen</p> <p>Opción 2: Migración de esquema con herramientas diseñadas para la base de datos de origen + AWS DMS para la carga de datos</p>	<p>Opción 1: Migración con herramientas diseñadas para la base de datos de origen + replicación de logs binarios</p> <p>Opción 2: Migración de esquema con herramientas diseñadas para la base de datos de origen + AWS DMS para transferir los datos</p>
Servidor Oracle/SQL	<p>Opción 1: Herramienta de conversión de esquemas de AWS + AWS DMS (recomendado)</p> <p>Opción 2: Conversión de esquema manual o mediante herramienta de terceros + carga de datos en destino manual o mediante herramienta de terceros</p>	<p>Opción 1: Herramienta de conversión de esquemas de AWS + AWS DMS (recomendado)</p> <p>Opción 2: Conversión de esquema manual o mediante herramienta de terceros + carga de datos en destino manual o mediante herramienta de terceros + replicación mediante herramienta de terceros</p>
Otras bases de datos distintas a MySQL	<p>Opción: Conversión de esquema manual o mediante herramienta de terceros + carga de datos en destino manual o mediante herramienta de terceros</p>	<p>Opción: Conversión de esquema manual o mediante herramienta de terceros + carga de datos en destino manual o mediante herramienta de terceros + replicación mediante herramienta de terceros (GoldenGate, etc.)</p>

*Herramientas diseñadas para MySQL: mysqldump, SELECT INTO OUTFILE, herramientas de terceros como mydumper/myloader

Migración de instantánea de RDS

Para utilizar la migración de instantánea de RDS al realizar la transición a Aurora, la base de datos MySQL se debe estar ejecutando en MySQL 5.6 de

Amazon RDS, y deberá realizar una instantánea RDS de la base de datos. Este método de migración no funciona con bases de datos en las instalaciones del cliente o con bases de datos que se ejecutan en Amazon Elastic Compute Cloud (Amazon EC2). Además, si su base de datos utiliza una versión de MySQL de Amazon RDS anterior a la 5.6, necesitará actualizarla a la 5.6 para poder comenzar.

La principal ventaja de este método de migración es que es el más sencillo y el que requiere un menor número de pasos. En particular, se encarga de migrar todos los objetos del esquema, los índices secundarios y los procedimientos almacenados junto con todos los datos de la base de datos.

Durante la migración instantánea sin replicación de logs binarios, la base de datos de origen tiene que estar sin conexión o en modo de solo lectura (a fin de que no se produzcan cambios en la base de datos de origen durante la migración). Para estimar el tiempo de inactividad, no tiene más que utilizar la instantánea de la base de datos existente para realizar una migración de prueba. Si el tiempo de inactividad de su migración se adapta a sus requisitos, este puede ser el método que más le convenga. Tenga en cuenta que, en algunos casos, la migración mediante AWS DMS o con herramientas de migración diseñadas para la base de datos de origen pueden ser más rápidos que la migración de instantánea.

Si no puede permitirse el tiempo de inactividad adicional, aún puede realizar una migración con un tiempo de inactividad prácticamente nulo si migra en primer lugar una instantánea de la base de datos de RDS a Amazon Aurora, al tiempo que permite que la base de datos de origen que se siga actualizando. Acto seguido, se encargará de actualizarla utilizando la replicación de logs binarios desde MySQL a Aurora.

Puede migrar una instantánea de base de datos automática o manual. Los pasos generales que debe seguir son los siguientes:

1. Determine la cantidad de espacio necesario para migrar las instancias de MySQL 5.6 de Amazon RDS a un clúster de base de datos de Aurora. Para obtener más información, consulte la siguiente sección.
2. Utilice la consola de Amazon RDS para crear la instantánea en la región en la que se encuentra la instancia de MySQL 5.6 de Amazon RDS.

3. Utilice la característica **Migrate Database** en la consola para crear un clúster de base de datos de Amazon Aurora que se llenará con la instantánea original de la base de datos de la instancia de MySQL 5.6.

Nota: Puede que algunas tablas MyISAM no se conviertan sin errores, por lo que pueden requerir modificaciones manuales. Por ejemplo, el motor de InnoDB no permite que un campo autoincremental forme parte de una clave compuesta. Además, en la actualidad no se admiten índices espaciales.

Cálculo de los requisitos de espacio para la migración de instantánea

Al migrar una instantánea de una instancia de base de datos MySQL a un clúster de base de datos de Aurora, esta última utiliza un volumen de Amazon Elastic Block Store (Amazon EBS) para formatear los datos de la instantánea antes de proceder a la migración. Existen algunos casos en los que se necesita espacio adicional para formatear los datos para la migración. Las dos características que pueden provocar problemas de espacio durante la migración son las tablas MyISAM y el uso de la opción `ROW_FORMAT=COMPRESSED`. Si no utiliza ninguna de estas características en su base de datos de origen, puede omitir esta sección, ya que no debería tener problemas de espacio. Durante la migración, las tablas MyISAM se convierten en tablas InnoDB, y todas ellas se descomprimen. Por lo tanto, tiene que calcularse un margen de espacio adecuado para las copias adicionales de dichas tablas.

El tamaño de la migración se basa en el tamaño asignado de la base de datos MySQL de origen de la que se hizo la instantánea. Por lo tanto, si usa tablas comprimidas o de tipo MyISAM que supongan un pequeño porcentaje del tamaño total de la base de datos, y dispone de espacio libre en la base de datos original, la migración debería llevarse a cabo correctamente sin encontrarse con problemas de espacio. Sin embargo, si la base de datos original no tiene suficiente espacio para almacenar una copia de las tablas MyISAM convertidas, así como otra copia (sin comprimir) de tablas comprimidas, el volumen para la migración no será suficiente. En esta situación, tendrá que modificar la base de datos MySQL de Amazon RDS para aumentar la asignación de tamaño de la base de datos con el objetivo de dejar espacio para las copias adicionales de estas tablas y, después, realizar una nueva instantánea de la base de datos y, finalmente, migrar la nueva instantánea.

Cuando realice la migración de datos a su clúster de la base de datos, tenga en cuenta las siguientes recomendaciones y limitaciones:

- Aunque Amazon Aurora admite hasta 64 TB de almacenamiento, el proceso de migrar una instantánea a un clúster de base de datos de Aurora está limitado por el tamaño del volumen de la instantánea de Amazon EBS y, por tanto, se limita a un tamaño máximo de 6 TB.
- Las tablas de la base de datos de origen que no son MyISAM pueden tener un tamaño de hasta 6 TB. No obstante, debido a los requisitos de espacio adicional durante la conversión, asegúrese de que ninguna de las tablas MyISAM o comprimidas que se vayan a migrar desde su instancia de la base de datos MySQL supere los 3 TB de tamaño.

Es posible que le interese modificar su esquema de base de datos (convertir las tablas MyISAM a InnoDB y eliminar `ROW_FORMAT=COMPRESSED`) antes de realizar la migración a Amazon Aurora. Esto puede ser útil en los siguientes casos:

- Desea acelerar el proceso de migración.
- No está seguro de la cantidad de espacio que necesita reservar.
- Ha intentado migrar los datos y la migración ha fallado debido a que no ha reservado suficiente espacio.

Asegúrese de que no está realizando estos cambios en la base de datos MySQL de Amazon RDS que utiliza para la producción, sino en una instancia de la base de datos que haya sido restaurada a partir de una instantánea de producción. Para obtener más información sobre cómo hacerlo, consulte [Reducing the Amount of Space Required to Migrate Data into Amazon Aurora](#) en el documento *Amazon RDS User Guide*⁵.

Migración de una instantánea de la base de datos utilizando la consola

Puede migrar una instantánea de la base de datos de una instancia MySQL de Amazon RDS para crear un clúster de la base de datos de Aurora. El nuevo clúster de la base de datos se rellena con los datos de la instancia original de la base de datos MySQL de Amazon RDS. La instantánea de la base de datos se debe haber obtenido a partir de una instancia de RDS con MySQL 5.6, y no debe estar cifrada. Para obtener información sobre la creación de una instantánea de la base de datos, consulte [Creating a DB Snapshot](#) en el documento *Amazon RDS User Guide*⁶.

Si la instantánea de la base de datos no se encuentra en la región en la que desea ubicar el clúster de la base de datos de Aurora, utilice la consola de Amazon RDS

para copiar la instantánea de la base de datos a esa región. Para obtener información acerca de copiar una instantánea de base de datos, consulte [Copying a DB Snapshot](#) en el documento *Amazon RDS User Guide*⁷.

Para migrar una instantánea de una base de datos MySQL 5.6 mediante la utilización de la consola, haga lo siguiente:

1. Inicie sesión en la consola de administración de AWS y abra la consola de Amazon RDS en <https://console.aws.amazon.com/rds/>.
2. Elija **Snapshots**.
3. En la página **Snapshots**, elija la instantánea de MySQL 5.6 de Amazon RDS que desea migrar a un clúster de base de datos de Aurora.
4. Elija **Migrate Database**.
5. En la página **Migrate Database**, especifique los valores que coincidan con sus requisitos de procesamiento y de entorno, tal y como se muestra en la siguiente ilustración. Para las descripciones de estas opciones, consulte [Migrating a DB Snapshot by Using the Console](#) en el documento *Amazon RDS User Guide*⁸.

Migrate Database

Migrate this database to a new DB Engine by selecting your desired options for the migrated instance.

Instance Specifications

Migrate to DB Engine:

DB Instance Class:

Settings

DB Snapshot ID: rds:ca-2015-01-mysql-maz-vpc-db-m3-medium-117154-ukbv-2015-06-10-00-01

DB Instance Identifier*: -mysql-maz-vpc-db-m3-medium-

Network & Security

This instance will be created with the new Certificate Authority rds-ca-2015. If you are using SSL to connect to this instance, you should use the [new certificate bundle](#). Learn more [here](#).

VPC*:

Subnet Group:

Publicly Accessible:

Availability Zone:

Database Options

Database Port:

Maintenance

Auto Minor Version Upgrade:

Figura 2: Migración de instantánea

- Haga clic en **Migrate** para migrar la instantánea de la base de datos.

En la lista de instancias, haga clic en el icono de flecha adecuado para mostrar los detalles del clúster de la base de datos y supervisar el avance de la migración. Este panel de detalles muestra el extremo del clúster principal que se utiliza para conectarse a la instancia principal del clúster de la base de datos. Para obtener

más información sobre la conexión a un clúster de base de datos de Amazon Aurora, consulte [Connecting to an Amazon Aurora DB Cluster](#) en el documento *Amazon RDS User Guide*⁹.

Migración del esquema de la base de datos

La migración de instantáneas de la base de datos de RDS transfiere tanto el esquema como los datos completos a la nueva instancia de Aurora. Sin embargo, si la ubicación de la base de datos de origen o los requisitos de tiempo de actividad de la aplicación no permiten usar la migración de instantánea de RDS, tendrá que migrar primero el esquema de la base de datos desde la base de datos de origen a la base de datos de destino antes de poder mover los datos reales. Un esquema de base de datos es un esqueleto lógico que representa la vista lógica de toda la base de datos, y normalmente incluye los siguientes elementos:

- Objetos de almacenamiento de la base de datos: tablas, columnas, restricciones, índices, secuencias, tipos definidos por el usuario y tipos de datos
- Objetos de código de la base de datos: funciones, procedimientos, paquetes, activadores, vistas, vistas materializadas, eventos, funciones escalares de SQL, funciones en línea de SQL, funciones de tabla SQL, atributos, variables, constantes, tipos de tabla, tipos públicos, tipos privados, cursores, excepciones, parámetros y otros objetos

En la mayoría de las situaciones, el esquema de base de datos permanece relativamente estático y, por ello, no es necesario aplicar un tiempo de inactividad durante la fase de migración de dicho esquema. El esquema de la base de datos de origen se puede obtener mientras la base de datos está activa sin que el rendimiento se vea afectado. Si la aplicación o los desarrolladores realizan cambios frecuentes en el esquema de la base de datos, asegúrese de que estos cambios se han detenido mientras se lleva a cabo la migración o bien son atendidos durante el proceso de migración del esquema.

En función del tipo de base de datos de origen, puede utilizar las técnicas que trataremos en las próximas secciones para migrar el esquema de la base de datos. Como requisito previo para la migración del esquema, debe disponer de una base de datos de Aurora ya creada y disponible.

Migración de esquema homogénea

Si la base de datos de origen es compatible con MySQL 5.6 y se ejecuta en Amazon RDS, Amazon EC2, o fuera de AWS, puede utilizar herramientas nativas de MySQL para exportar e importar el esquema.

- **Exportación del esquema de base de datos.** Puede utilizar la aplicación de cliente [mysqldump](#) para exportar el esquema de la base de datos. Para ejecutar esta aplicación, debe conectarse a la base de datos de origen y dirigir los archivos resultantes del comando `mysqldump` a un archivo plano. La opción `--no-data` garantiza que se exporte solamente el esquema de la base de datos, excluyendo los datos de tablas. Puede leer la documentación completa sobre comandos `mysqldump` en [mysqldump—A Database Backup Program](#)¹⁰.

```
mysqldump -u source_db_username -p --no-data --routines --triggers -databases source_db_name > DBSchema.sql
```

- **Importación del esquema de la base de datos en Aurora.** Para importar el esquema a su instancia de Aurora, conéctese a la base de datos de Aurora desde un cliente de línea de comandos de MySQL (o un cliente correspondiente de Windows) y dirija el contenido del archivo de exportación a MySQL.

```
mysql -h aurora-cluster-endpoint -u username -p <DBSchema.sql
```

Tenga en cuenta lo siguiente:

- Si la base de datos de origen contiene vistas, procedimientos y activadores almacenados, tendrá que eliminar la sintaxis `DEFINER` de su archivo de volcado. A continuación se ofrece un sencillo comando de Perl para llevarlo a cabo. Este procedimiento crea todos los activadores, las vistas y los procedimientos almacenados con el usuario actualmente conectado como `DEFINER`. Asegúrese de evaluar las posibles implicaciones de seguridad.

```
$perl -pe 's/\sDEFINER=[^`]+@[^`]+`// ' < DBSchema.sql >  
DBSchemaWithoutDEFINER.sql
```

- Amazon Aurora es compatible únicamente con tablas InnoDB. Si su base de datos de origen posee tablas MyISAM, Aurora cambiará automáticamente el motor a InnoDB cuando se ejecute el comando `CREATE TABLE`.
- Amazon Aurora no es compatible con tablas comprimidas (es decir, tablas creadas con `ROW_FORMAT=COMPRESSED`). Si su base de datos de origen posee tablas comprimidas, Aurora pasará automáticamente `ROW_FORMAT` a `COMPACT` cuando se ejecute el comando `CREATE TABLE`.

Una vez haya importado correctamente el esquema a Amazon Aurora desde la base de datos de origen compatible con MySQL 5.6, el siguiente paso será copiar los datos reales de la fuente en el destino. Para obtener más información, consulte más adelante, en este documento, la guía [Introducción y enfoque general para AWS DMS](#).

Migración de esquema heterogénea

Si la base de datos de origen no es compatible con MySQL, deberá convertir su esquema a un formato compatible con Amazon Aurora. La conversión de esquema de un motor de base de datos a otro es una tarea muy importante, y puede conllevar la reescritura de determinadas partes de la base de datos y del código de las aplicaciones. Tiene dos opciones principales para convertir y migrar su esquema a Amazon Aurora:

- **Herramienta de conversión de esquema de AWS.** La [herramienta de conversión de esquemas de AWS](#) facilita la migración de bases de datos heterogéneas convirtiendo de forma automática el esquema de la base de datos de origen y la mayor parte del código personalizado, incluyendo vistas, procedimientos almacenados y funciones, a un formato compatible con la base de datos de destino¹¹. Cualquier código que no se pueda convertir automáticamente se marca con claridad para que se pueda convertir de forma manual. Puede utilizar esta herramienta para convertir las bases de datos de origen que se ejecutan en Oracle o Microsoft SQL Server a una base de datos de destino de Amazon Aurora, MySQL o PostgreSQL tanto en Amazon RDS como en Amazon EC2. El método

preferido consiste en usar la herramienta de conversión de esquemas de AWS para convertir el esquema de Oracle, SQL Server o PostgreSQL esquema a un formato compatible con Aurora.

- **Migración de esquema manual y herramientas de terceros.** Si la base de datos de origen no es Oracle, SQL Server o PostgreSQL, puede migrar manualmente el esquema de la base de datos de origen a Aurora o bien utilizar herramientas de terceros para migrar el esquema en un formato que sea compatible con MySQL 5.6. La migración de esquema manual puede ser un proceso bastante complicado, dependiendo del tamaño y la complejidad de su esquema de origen. En la mayoría de los casos, sin embargo, merece la pena realizar la conversión de esquema manual por el ahorro de costos, las ventajas de rendimiento y otras mejoras que se incluyen con Amazon Aurora.

Migración de esquema con la herramienta de conversión de esquemas de AWS

La herramienta de conversión de esquemas de AWS proporciona una interfaz de usuario basada en proyectos diseñada para convertir automáticamente el esquema de la base de datos de origen en un formato compatible con Amazon Aurora. Es muy recomendable que utilice la herramienta de conversión de esquemas de AWS para evaluar el esfuerzo requerido para la migración de la base de datos y para realizar una migración piloto antes de abordar la migración de producción real.

La siguiente descripción le muestra los pasos principales que se deben seguir al utilizar de la herramienta de conversión de esquemas de AWS. Para obtener instrucciones detalladas, consulte la sección [Getting Started](#) del documento *AWS Schema Conversion Tool User Guide*¹².

1. En primer lugar, instale la herramienta. La herramienta de conversión de esquemas de AWS está disponible para los motores de base de datos Microsoft Windows, Mac OS X, Ubuntu Linux y Fedora Linux.

Puede encontrar las instrucciones detalladas de descarga e instalación en la sección [Installation and update](#) de la guía de usuario¹³. La elección del lugar en el que instalar la herramienta de conversión de esquemas de AWS es importante. La herramienta debe conectarse directamente a las bases de datos de origen y de destino con el fin de convertir y aplicar el esquema. Asegúrese de que el escritorio en el que instale la herramienta de

conversión de esquemas de AWS dispone de conectividad de red con las bases de datos de origen y de destino.

2. Instalar los controladores JDBC. La herramienta de conversión de esquemas de AWS utiliza controladores JDBC para conectarse a las bases de datos de origen y de destino. Para utilizar esta herramienta, debe descargar dichos controladores JDBC a su escritorio local. Las instrucciones para la descarga de los controladores se encuentran en [Required Database Drivers](#) en el documento *AWS Schema Conversion Tool User Guide*¹⁴. Asimismo, consulte el [AWS forum for AWS Schema Conversion Tool](#) para obtener instrucciones sobre cómo configurar JDBC para diferentes motores de bases de datos¹⁵.
3. Creación de una base de datos de destino. Creación de una base de datos de destino de Amazon Aurora. Para obtener instrucciones sobre cómo crear una base de datos de Amazon Aurora, consulte [Creating an Amazon Aurora DB Cluster](#) en el documento *Amazon RDS User Guide*¹⁶.
4. Abra la herramienta de conversión de esquemas de AWS y comience a utilizar el **New Project Wizard**.

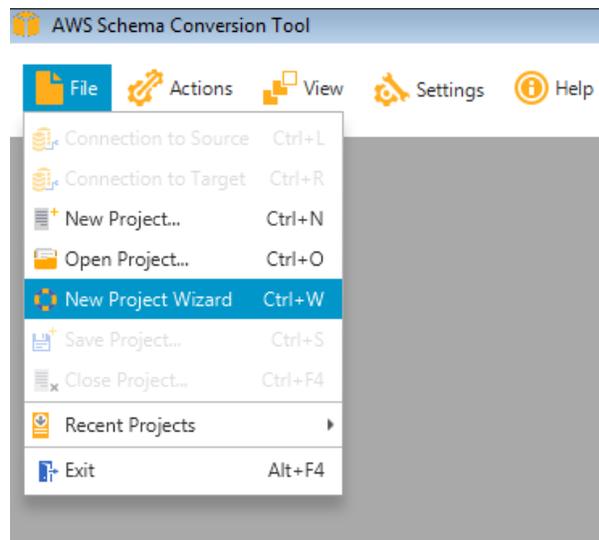


Figura 3: Creación de un nuevo proyecto en la herramienta de conversión de esquemas de AWS

5. Configure la base de datos de origen y compruebe la conectividad entre la herramienta de conversión de esquemas de AWS y la base de datos de origen. La base de datos de origen debe ser accesible desde el escritorio

para que funcione, así que asegúrese de que ha configurado correctamente la red y el cortafuegos.

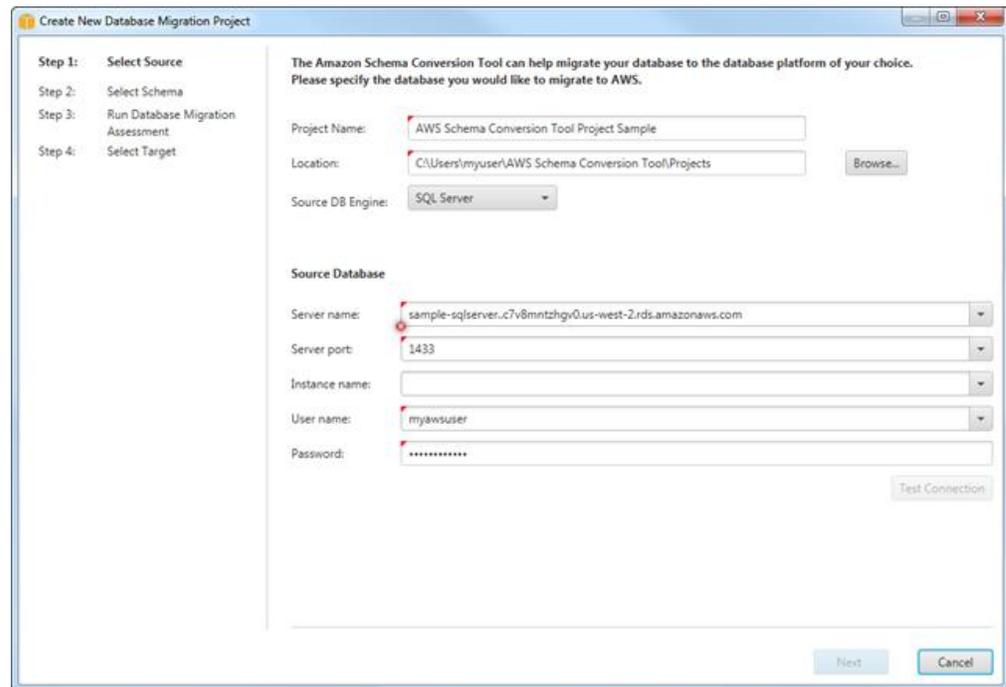


Figura 4: Asistente para la creación de un nuevo proyecto de migración de bases de datos

6. En la siguiente pantalla, seleccione el esquema de la base de datos de origen que desea convertir a Amazon Aurora.

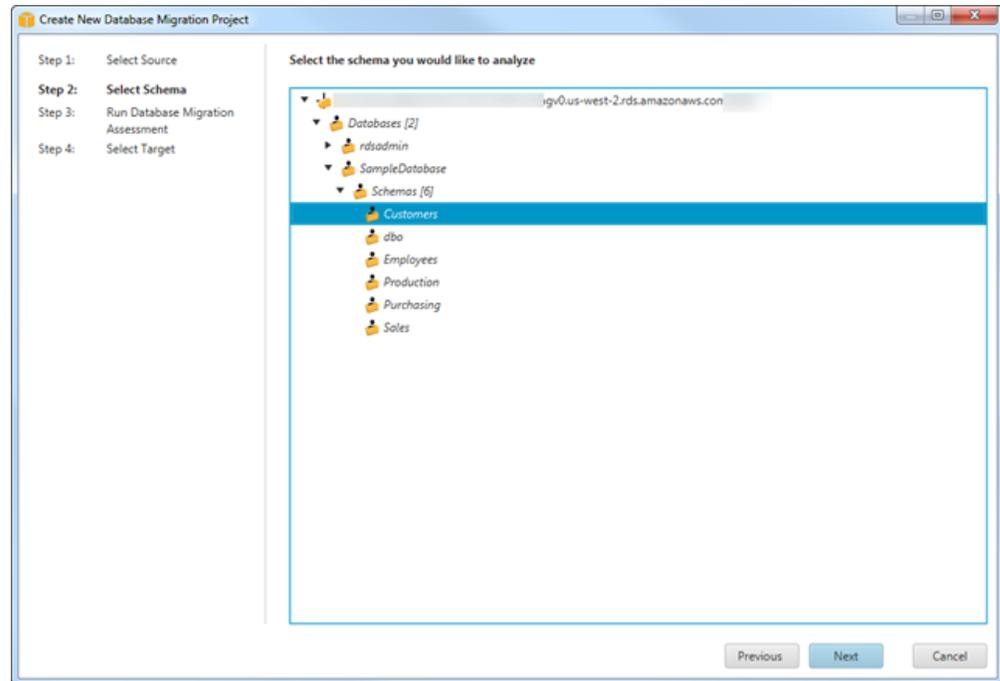


Figura 5: Selección de esquema en el asistente de migración

7. Ejecute el informe de evaluación de la migración de la base de datos. Este informe proporciona información importante acerca de la conversión del esquema de la base de datos de origen a la instancia de destino de Amazon Aurora. Dicho informe resume todas las tareas de conversión del esquema e indica, de forma detallada, las acciones pendientes que se deben realizar para aquellas partes del esquema que no se pueden convertir automáticamente a Aurora. El informe también incluye estimaciones para el tiempo que requerirá escribir el código equivalente que no se ha podido convertir de forma automática en la base de datos de destino.

Haga clic en **Next** para configurar la base de datos de destino. Puede ver este informe de migración de nuevo más tarde.

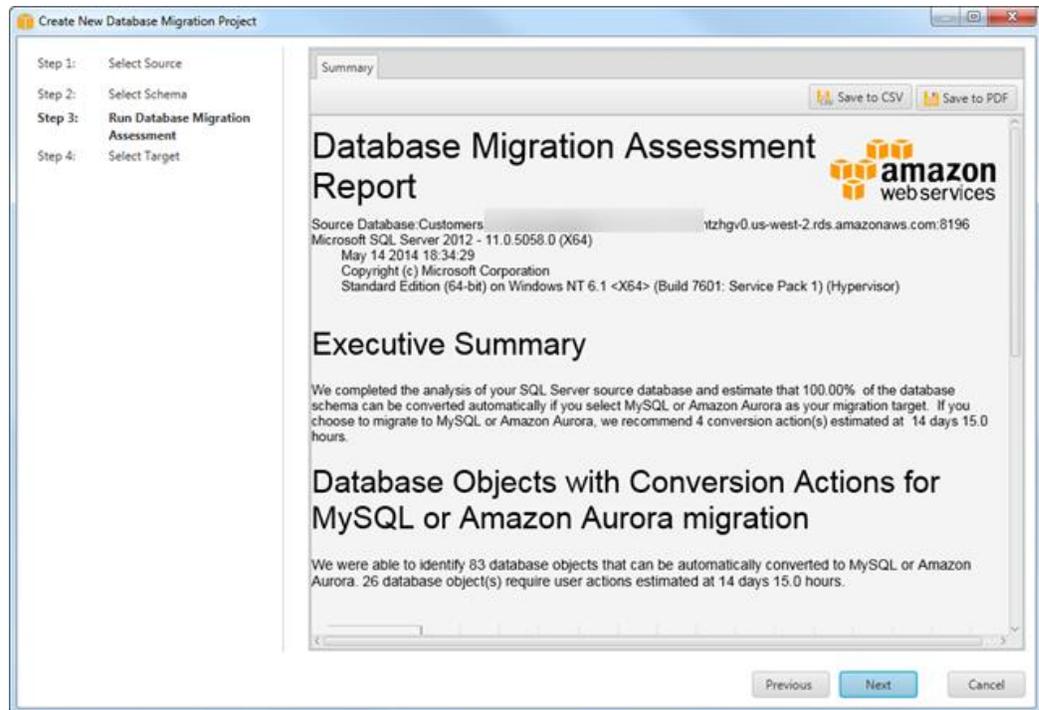


Figura 6: Informe de migración

8. Configure la base de datos de Amazon Aurora de destino y compruebe la conectividad entre la herramienta de conversión de esquemas de AWS y la base de datos de origen. La base de datos de destino debe ser accesible desde el escritorio para que funcione, así que asegúrese de que ha configurado correctamente la red y el cortafuegos. Haga clic en **Finish** para ir a la ventana de proyecto.
9. Cuando esté en la ventana de proyecto, ya habrá establecido una conexión a las bases de datos de origen y de destino, y estará listo para estudiar el informe de evaluación detallado y migrar el esquema.
10. En el panel izquierdo que indica el esquema de la base de datos de origen, elija un objeto del esquema para crear un informe de evaluación. Haga clic con el botón derecho en el objeto y elija **Create Report**.

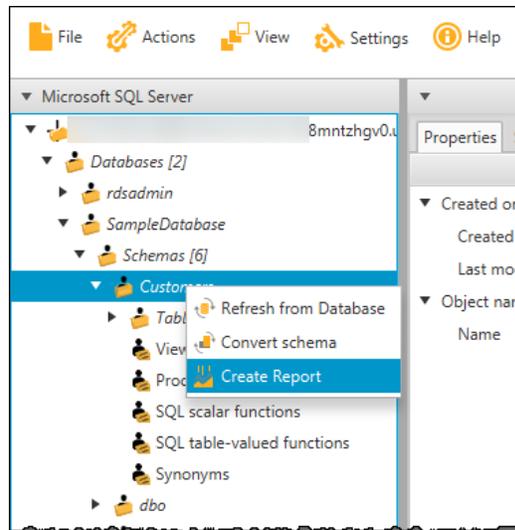


Figura 7: Creación del informe de migración

La pestaña **Summary** muestra la información resumida en el informe de evaluación de la migración de la base de datos. Muestra los elementos que se han convertido automáticamente y los que no se han podido convertir.

Para aquellos elementos del esquema que no se han podido convertir automáticamente al motor de la base de datos de destino, el resumen incluye una estimación del tiempo que le llevaría crear un esquema equivalente al de a su base de datos de origen en la instancia de la base de datos de destino. El informe clasifica el tiempo estimado para convertir estos elementos del esquema de la siguiente forma:

- **Simple** - Acciones que se pueda completar en menos de 1 hora.
- **Medium** - Acciones que son más complejas y se pueden realizar en un plazo de una a cuatro horas.
- **Significant** - Acciones que son muy complejas y que tardarán más de cuatro horas en completarse.

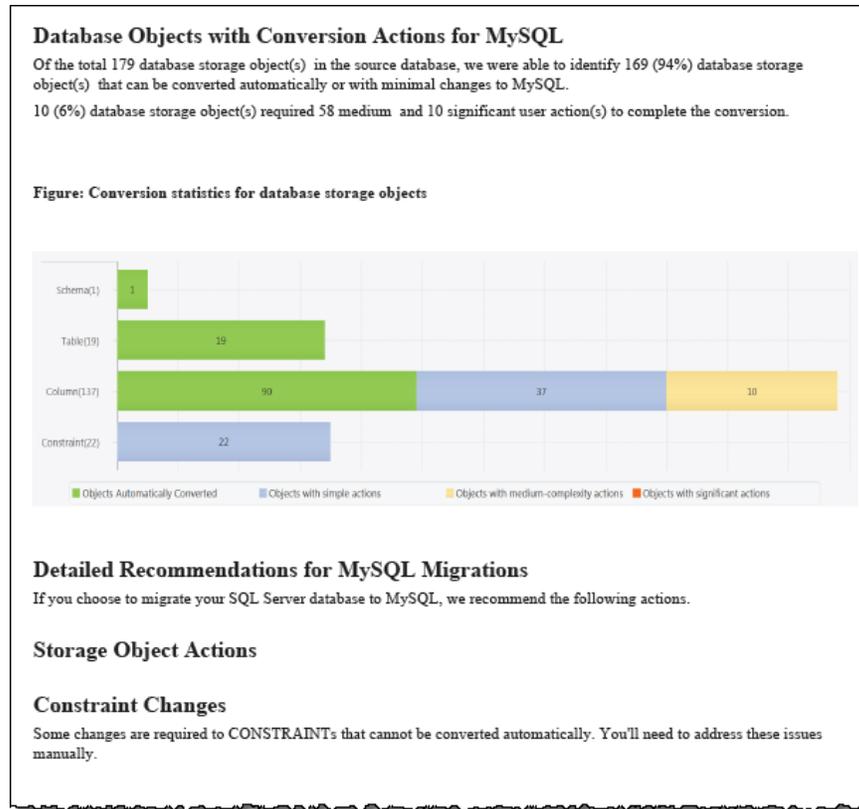


Figura 8: Informe de migración

Importante: Si está evaluando el esfuerzo necesario para llevar a cabo su proyecto de migración de la base de datos, debería considerar este informe de evaluación como un importante instrumento. Estudie el informe de evaluación con todo detalle para determinar qué partes del código se deben modificar en el esquema de la base de datos y qué impacto podrían tener estos cambios en la funcionalidad y diseño de la aplicación.

- El siguiente paso consiste en convertir el esquema. El esquema convertido no se aplica de forma inmediata a la base de datos de destino. En su lugar, se almacena localmente hasta que el esquema convertido se aplique de forma explícita a la base de datos de destino. Para convertir el esquema de la base de datos de origen, elija un objeto del esquema que desee convertir desde el panel izquierdo del proyecto. Haga clic con el botón derecho en el objeto y elija **Convert schema**, tal y como se muestra en la siguiente ilustración.

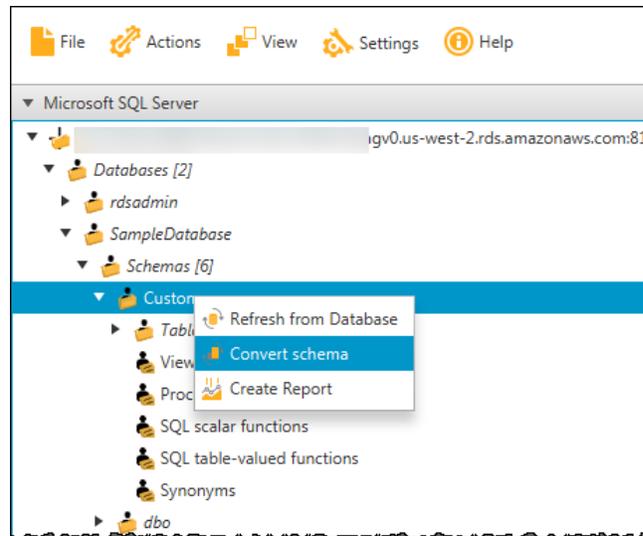


Figura 9: Conversión del esquema

Esta acción añade el esquema convertido al panel derecho de la ventana de proyectos, y muestra los objetos que se han convertido automáticamente mediante la herramienta de conversión de esquemas de AWS.

12. Puede responder a las acciones pendientes indicadas en el informe de evaluación de diferentes formas:
 - Añada el esquema equivalente de forma manual. Puede escribir la parte del esquema que se puede convertir automáticamente a la instancia de la base de datos de destino escogiendo en el panel derecho de su proyecto **Apply to database**. El esquema que se registra en la instancia de la base de datos de destino no contiene los elementos que no se han podido convertir automáticamente. Estos elementos se indican en el informe de evaluación de la migración de la base de datos.

Después de aplicar el esquema a la instancia de la base de datos de destino, puede crear manualmente dicho esquema para aquellos elementos que no se han podido convertir automáticamente. En algunos casos, es posible que no pueda crear un esquema equivalente en la instancia de la base de datos de destino. Es posible que tenga que volver a diseñar una parte de la aplicación y de la base de datos para utilizar la funcionalidad que está disponible desde el motor de base de datos para la instancia de la base de datos de destino. En otros casos, puede ignorar sin más el esquema que no se pueda convertir automáticamente.

Precaución: Si decide crear manualmente el esquema de la instancia de la base de datos de destino, no elija **Apply to database** hasta que no haya guardado una copia de todo lo que haya hecho a mano. Cuando aplica el esquema de su proyecto a la instancia de la base de datos de destino, este sobrescribe el esquema del mismo nombre en la instancia de la base de datos de destino, con lo que perderá las modificaciones que haya añadido a mano con posterioridad.

- Modifique el esquema de la base de datos de origen y actualice el esquema en su proyecto. Para algunos elementos, puede que la elección más eficaz sea modificar el esquema de la base de datos en la base de datos de origen, adaptándolo al esquema que sea compatible con la arquitectura de su aplicación y que, además, se pueda convertir automáticamente al motor de la base de datos de la instancia de destino. Después de actualizar el esquema de la base de datos de origen y de comprobar que las actualizaciones son compatibles con su aplicación, elija **Refresh from Database** en el panel izquierdo del proyecto para actualizar el esquema de la base de datos de origen. A continuación, puede convertir su esquema actualizado y generar de nuevo el informe de evaluación de la migración de la base de datos. Verá que la acción pendiente para su esquema actualizado ya no aparece.
13. Cuando esté listo para aplicar el esquema convertido a la instancia de Aurora de destino, elija el elemento del esquema desde el panel derecho de su proyecto. Haga clic con el botón derecho en el elemento del esquema de la base de datos y elija **Apply to database**, tal y como se muestra en la siguiente ilustración.

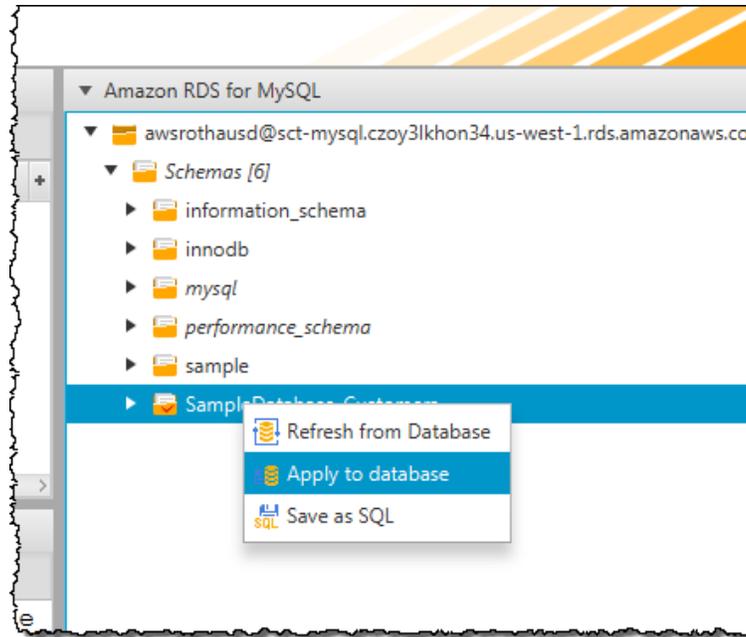


Figura 10: Aplicación del esquema a la base de datos

Nota: La primera vez que aplica el esquema convertido a su instancia de la base de datos de destino, la herramienta de conversión de esquemas de AWS añade un esquema adicional (`AWS_ORACLE_EXT` o `AWS_SQLSERVER_EXT`) a la instancia de la base de datos de destino. Este esquema implementa las funciones del sistema de la base de datos de origen que se necesitan a la hora de escribir el esquema convertido a la instancia de la base de datos de destino. No modifique este esquema; de lo contrario, es posible que obtenga resultados inesperados en el esquema convertido que se escriba en la instancia de la base de datos de destino. Cuando el esquema se haya migrado por completo a la instancia de la base de datos de destino, y ya no necesite la herramienta de conversión de esquemas de AWS, puede eliminar el esquema `AWS_ORACLE_EXT` o `AWS_SQLSERVER_EXT`.

La herramienta de conversión de esquemas de AWS representa una opción extra fácil de usar para el kit de herramientas de migración. Para conocer otras prácticas recomendadas relacionadas con la herramienta de conversión de esquemas de AWS, consulte la sección [Best Practices](#) en el documento *AWS Schema Conversion Tool User Guide*¹⁷.

Migración de datos

Después de haber copiado el esquema de la base de datos de origen a la base de datos de Aurora de destino, el siguiente paso consiste en migrar los datos reales. Aunque la migración de datos se puede llevar a cabo con diferentes herramientas, le recomendamos utilizar AWS Database Migration Service (AWS DMS), ya que ofrece la simplicidad y las características necesarias para completar la tarea que tiene delante.

Introducción y enfoque general para AWS DMS

AWS Database Migration Service (AWS DMS) facilita a los clientes la tarea de migrar bases de datos de producción a AWS con un tiempo de inactividad mínimo. Puede seguir ejecutando las aplicaciones durante la migración de la base de datos. Además, AWS Database Migration Service garantiza que los cambios en la base de datos de origen que se realizan durante y después de la migración se repliquen de forma ininterrumpida en la base de datos de destino. Las tareas de migración se pueden configurar en cuestión de minutos en la consola de administración de AWS. AWS Database Migration Service puede migrar los datos desde o hacia varias de las plataformas de bases de datos más populares como, por ejemplo, Oracle, SQL Server, MySQL, PostgreSQL, Amazon Aurora, MariaDB y Amazon Redshift.

El servicio admite migraciones homogéneas, como de Oracle a Oracle, además de migraciones heterogéneas entre diferentes plataformas de base de datos, como de Oracle a Amazon Aurora o de SQL Server a MySQL. Puede realizar migraciones de una sola vez, o puede mantener la replicación continua entre las bases de datos sin que el cliente tenga que instalar o configurar ningún programa complicado.

AWS DMS funciona con bases de datos que se encuentran en las instalaciones y que se ejecutan en Amazon EC2, o bien que se ejecutan en Amazon RDS. Sin embargo, AWS DMS no funciona en situaciones en las que tanto la base de datos de origen como la de destino se encuentran en las instalaciones, ya que uno de los extremos debe encontrarse en AWS.

AWS DMS es compatible con versiones concretas de Oracle, SQL Server, Amazon Aurora, MySQL y PostgreSQL. Para conocer las versiones compatibles, consulte el documento [AWS Database Migration Service User Guide](#)¹⁸. Sin embargo, este

documento técnico se centra únicamente en Amazon Aurora como destino de la migración.

Métodos de migración

AWS DMS ofrece tres métodos a la hora de migrar sus datos:

Migración de los datos existentes. Este método crea las tablas en la base de datos de destino, donde define automáticamente los metadatos necesarios y rellena las tablas con los datos de la base de datos de origen (método también denominado "carga completa"). Los datos de las tablas se cargan en paralelo para mejorar la eficacia. Las tablas solo se crean en caso de migraciones homogéneas, y AWS DMS no crea los índices secundarios de forma automática. Siga leyendo para obtener más información.

Migración de datos existentes y replicación de cambios en curso. Este método admite una carga completa, tal y como se describe anteriormente y, además, captura las modificaciones que se estén realizando en la base de datos de origen durante la carga completa y los almacena en la instancia de replicación. Tras haber completado la carga completa, se aplican los cambios almacenados en la base de datos de destino hasta que esté completamente actualizada respecto a la base de datos de origen. Asimismo, los cambios en curso de la base de datos de origen se siguen replicando a la base de datos de destino para mantener ambas sincronizadas. Este método de migración es muy útil cuando desea realizar una migración de la base de datos con un tiempo de inactividad muy reducido.

Replicación de los cambios en los datos únicamente. Este método solo lee los cambios en el archivo de registro de recuperación de la base de datos de origen y los aplica a la base de datos de destino sobre la marcha. Si la base de datos de destino no está disponible, estos cambios se mantienen en la memoria intermedia de la instancia de replicación hasta que la base de datos de destino vuelva a estar disponible.

Cuando AWS DMS realiza una migración de carga completa, el procesamiento aplica una carga a las tablas de la base de datos de origen, lo que puede influir en el rendimiento de las aplicaciones que están sirviéndose de esta base de datos en ese momento. Si esto constituye un problema, y no puede cerrar sus aplicaciones durante la migración, puede considerar actuar de alguna de las siguientes formas:

- Llevar a cabo la migración en un momento en el que la carga de aplicaciones en la base de datos se encuentra en su punto más bajo.
- Crear una réplica de lectura de la base de datos de origen y, a continuación, realizar la migración con AWS DMS a partir de la réplica de lectura.

Procedimiento de migración

Los pasos generales para utilizar AWS DMS son los siguientes:

1. Cree una base de datos de destino.
2. Copie el esquema.
3. Cree una instancia de replicación de AWS DMS.
4. Defina los extremos de la base de datos de origen y de destino.
5. Cree y ejecute la tarea de migración.

Creación de los datos de destino

Cree su clúster de la base de datos de destino de Amazon Aurora utilizando el procedimiento descrito en [Creating an Amazon Aurora DB Cluster](#) en el documento *Amazon RDS User Guide*¹⁹. Debe crear la base de datos de destino en la región y con un tipo de instancia que se adapte a sus requisitos de negocio. Además, para mejorar el rendimiento de la migración, compruebe que la base de datos de destino no tiene habilitada la implementación en zonas de disponibilidad múltiples (Multi-AZ); podrá habilitarla una vez haya finalizado la carga.

Copia del esquema

Además, debe crear el esquema en la base de datos de destino. AWS DMS admite la migración básica de esquema, que incluye la creación de tablas y las claves principales. Sin embargo, AWS DMS no creará automáticamente índices secundarios, claves externas, procedimientos almacenados, cuentas de usuario, etc., en la base de datos de destino. Para más información, consulte la sección [Migración del esquema de la base de datos](#).

Creación de una instancia de replicación de AWS DMS

Para utilizar el servicio AWS DMS, debe crear una instancia de replicación de AWS DMS, que se ejecuta en su VPC. Esta instancia lee los datos de la base de datos de origen, se encarga de realizar las asignaciones de tabla especificadas

y escribe los datos en la base de datos de destino. En general, la migración de la base de datos se agiliza si se usa una instancia de replicación de mayor tamaño, si bien la migración también puede verse condicionada por otros factores como, por ejemplo, la capacidad de la conexión de las bases de datos de origen y de destino, la latencia, etc. Además, puede detener la instancia de replicación una vez haya finalizado la migración de la base de datos.



Figura 11: AWS Database Migration Service

En la actualidad, AWS DMS es compatible con las clases de instancia T2 y C4 para instancias de replicación. Las clases de instancia T2 son un nuevo tipo de instancia estándar de bajo costo diseñadas para proporcionar un nivel básico de desempeño de la CPU con la posibilidad asumir cargas variables por encima del nivel básico. Son aptas para desarrollar, configurar y probar el proceso de migración de la base de datos, así como para realizar tareas periódicas de migración de datos que se puedan beneficiar de la capacidad de la CPU para asumir cargas de trabajo variables. Las clases de instancia C4 están diseñadas para proporcionar el máximo desempeño del procesador y alcanzar una velocidad gestión de paquetes por segundo (PPS) significativamente superior, además de reducir la latencia y su fluctuación en la red. Si planea migrar bases de datos de gran tamaño y desea minimizar el tiempo para completar la operación, debería utilizar las clases de instancia C4.

Normalmente, la carga completa no necesita una capacidad de almacenamiento significativa en la instancia de replicación de AWS DMS. Sin embargo, si está ejecutando la replicación junto con la carga completa, los cambios realizados en la base de datos de origen se almacenarán en la instancia de replicación de AWS DMS mientras se realiza la carga completa. Por lo tanto, si está realizando la migración de una base de datos de gran tamaño que está recibiendo una gran cantidad de actualizaciones mientras dicha migración está en curso, podría consumir un volumen importante de almacenamiento en la instancia. El tipo de instancia C4 incluye 100 GB de almacenamiento, mientras que la clase T2 incluye

50 GB. Normalmente, estos volúmenes de almacenamiento deberían ser más que suficientes para la mayoría de migraciones.

Además, en algunos casos extremos en los que se migran bases de datos de gran tamaño con niveles de actividad muy elevados y con la replicación habilitada, es posible que la replicación de AWS DMS no consiga replicar los datos en tiempo real. Si detecta esta situación, puede que necesite detener los cambios en la base de datos de origen durante unos minutos para que la replicación asuma todos los cambios antes de volver a apuntar la aplicación a la base de datos de destino de Aurora.

Create replication instance

A replication instance initiates the connection between the source and target databases, transfers the data, and caches any changes that occur on the source database during the initial data load. Use the fields below to configure the parameters of your new replication instance including network and security information, encryption details, and performance characteristics.

The screenshot shows the 'Create replication instance' form in the AWS DMS console. The form has the following fields and values:

- Name:** replication-instance-1
- Description:** ReplicationInstance1 description
- Instance class:** dms.t2.medium
- VPC:** - Select One -
- Publicly accessible:**

At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Next'. Below the main form, there is a section for 'Advanced' options, which is currently collapsed.

Figura 12: Página de creación de la instancia de replicación en la consola de AWS DMS

Definición de extremos de las bases de datos de origen y de destino

La instancia de replicación utiliza un extremo para conectarse a una base de datos. Para realizar una migración de la base de datos, debe crear un extremo *tanto* en la base de datos de origen como en la de destino. Los extremos de la base de datos pueden alojarse en las instalaciones, en Amazon EC2, o en Amazon RDS, pero las bases de datos de origen y de destino no pueden encontrarse en las instalaciones al mismo tiempo.

Es muy recomendable probar la conexión del extremo de la base de datos después de definirlo. Tal y como se explica en este documento, puede realizar esta comprobación en la misma página que ha usado para crear el extremo de la base de datos.

Nota: Si su esquema de origen incluye limitaciones para claves externas, al crear el extremo de destino necesitará introducir lo siguiente para **Extra connection attributes** en la sección **Advanced**:

```
initstmt=SET FOREIGN_KEY_CHECKS=0
```

Esto deshabilita las comprobaciones de las claves externas mientras se van cargando las tablas de la base de datos de destino. Esto, a su vez, impide que la carga se interrumpa por culpa de comprobaciones fallidas de claves externas en tablas parcialmente cargadas.

Create database endpoint

A database endpoint is used by the replication server to connect to a database. The database specified in the endpoint can be on-premise, on RDS, in EC2 or in the cloud. Details should be specified in the form below. It is recommended that you test your endpoint connections here to avoid errors during processing.

Endpoint type Source Target ⓘ

Endpoint Identifier ⓘ

Endpoint Engine ⓘ

Server address

Port

User name

Password

▶ Advanced

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC

Replication instance

Refresh schemas after successful connection test ⓘ

Figura 13: Página de creación de extremos de la base de datos en la consola de AWS DMS

Creación y ejecución de una tarea de migración

Ahora que ha creado y probado los extremos de la base de datos de origen y de destino, puede crear una tarea para realizar la migración de datos. Al crear una tarea, debe especificar la instancia de replicación que ha creado, el tipo de método de migración de la base de datos (como se ha tratado anteriormente), el extremo de la base de datos de origen y el de destino para el clúster de la base de datos de Amazon Aurora.

Además, en **Task Settings**, si ya ha creado el esquema completo de la base de datos de destino, deberá cambiar el parámetro **Target table preparation mode** a **Do Nothing** en lugar de usar el valor predeterminado **Drop tables on target**. Este último valor podría provocar que pierda algunas partes de la definición de su esquema al como, por ejemplo, las limitaciones de claves externas al soltar y recrear tablas.

Al crear una tarea, puede crear asignaciones de tablas que especifiquen el esquema de origen junto con las tablas correspondientes que se deben migran al extremo de destino. El método de asignación predeterminado migra todas las tablas de origen a las tablas de destino con el mismo nombre si estas existen. De lo contrario, se crea(n) la(s) tabla(s) de origen en el destino, dependiendo de la configuración de su tarea. Además, también puede crear asignaciones personalizadas (utilizando un archivo JSON) si desea migrar únicamente determinadas tablas o si desea tener más control del proceso de asignación de tablas y campos. También puede optar por migrar solo un esquema o todos los esquemas del extremo de origen.

Create task

A task can contain one or more table mappings which define what data is moved from the source to the target. If a table does not exist on the target, it can be created automatically.

Task name

Replication instance

Source endpoint

Target endpoint

Migration type

Start task on create

Task Settings

Table mappings

Mapping method Default Custom

Schemas Single schema All schemas

Schema to migrate

DMS will create the schema on the target if it does not already exist

[Show JSON](#)

[Cancel](#) [Create task](#)

Figura 14: Página de creación de tareas en la consola de AWS DMS

Puede utilizar la consola de administración de AWS para supervisar el progreso de las tareas de AWS Database Migration Service (AWS DMS). También puede supervisar los recursos y la conectividad de red que se están utilizando. La consola muestra estadísticas básicas de AWS DMS para cada tarea, incluyendo el estado de las tareas, el porcentaje del progreso, el tiempo transcurrido y las estadísticas de las tablas, tal y como se muestra en la siguiente imagen.

Además, puede seleccionar una tarea y visualizar las métricas de rendimiento de dicha tarea, incluyendo la velocidad de transferencia, el número de registros migrados por segundo, el uso de disco y de memoria y la latencia.

Filter:

ID	Status	Source	Target	Type	Complete %	Elapsed time	Tables loaded	Tables loading	Tables queued
<input type="checkbox"/> migrate-rdsmysql-to-rdsauo	Load complete	rds-mysql-test-	aur-trg-02	Full Load	<div style="width: 100%;"><div style="width: 100%;"></div></div> 100	0m	10	0	0

Figura 15: Estado de las tareas en la consola de AWS DMS

Pruebas y puesta en servicio

Una vez se haya completado con éxito la migración del esquema y de los datos desde la base de datos de origen a Amazon Aurora, estará listo para realizar las pruebas completas de extremo a extremo de su proceso de migración. La forma de realizar las pruebas se debe ir perfeccionando después de cada migración de prueba, y el plan de migración final tiene que incluir un plan que garantice que se lleven a cabo las pruebas adecuadas en la base de datos migrada.

Pruebas de migración

Categoría de prueba	Finalidad
Pruebas de aceptación básicas	<p>Estas pruebas de migración previas a la puesta en servicio deben ejecutarse de forma automática al finalizar el proceso de migración de datos. Su objetivo principal consiste en verificar si la migración de datos se ha realizado correctamente. A continuación se muestran algunos de los resultados más comunes que ofrecen estas pruebas:</p> <ul style="list-style-type: none"> • Número total de elementos procesados • Número total de elementos importados • Número total de elementos omitidos • Número total de advertencias • Número total de errores <p>Si cualquiera de estos totales que arrojan las pruebas se desvía de los valores esperados, se deduce que la migración no se ha realizado correctamente, y deberá resolver los problemas antes de pasar al siguiente paso en el proceso o a la siguiente ronda de pruebas.</p>
Pruebas de funcionamiento	<p>Estas pruebas posteriores a la puesta en servicio comprueban la funcionalidad de la aplicación o aplicaciones que utilizan Aurora para almacenar sus datos. Incluyen una combinación de pruebas automatizadas y manuales. El objetivo principal de las pruebas funcionales consiste en identificar problemas en la aplicación provocados por la migración de datos a Aurora.</p>
Pruebas no funcionales	<p>Estas pruebas de migración posteriores a la puesta en servicio evalúan las características no funcionales de la aplicación como, por ejemplo, el rendimiento bajo niveles variables de carga.</p>
Pruebas de aceptación de usuario	<p>Estas pruebas posteriores a la puesta en servicio deben ser ejecutadas por los usuarios finales de la aplicación una vez que la migración de datos y la puesta en servicio se hayan completado. El objetivo de estas pruebas es que los usuarios finales decidan si la aplicación es lo suficientemente utilizable para cumplir con su función principal en la organización.</p>

Puesta en servicio

Cuando haya completado la migración final y las pruebas, deberá dirigir la aplicación a la base de datos de Amazon Aurora. Esta fase de la migración se conoce como *puesta en servicio*. Si la planificación y la fase de pruebas se han llevado a cabo correctamente, la migración no debería ocasionar problemas inesperados.

Acciones anteriores a la puesta en servicio

- Defina un margen de tiempo para la puesta en servicio: Identifique un espacio de tiempo en el que pueda realizar la migración a la nueva base de datos creando las mínimas interferencias posibles al negocio. Normalmente se suele elegir un período de baja actividad para la base de datos (por lo general, por la noche y durante los fines de semana).
- Asegúrese de que los cambios más recientes se han copiado: Si se ha adoptado un enfoque de migración con un tiempo de inactividad prácticamente nulo a la hora de replicar los cambios de la base de datos de origen en la de destino, asegúrese de que todos los cambios recientes de la base de datos de origen se han copiado, y que la base de datos de destino no está asimilando los cambios con un retraso significativo respecto a la de origen.
- Prepare secuencias de comandos para llevar a cabo los cambios en la configuración de la aplicación: Para lograr una correcta puesta en servicio, debe modificar los detalles de conexión de la base de datos en los archivos de configuración de la aplicación. Puede que las aplicaciones complejas y de gran tamaño necesiten actualizar los detalles de conexión en varios lugares. Asegúrese de que dispone de las secuencias de comandos para actualizar la configuración de la conexión de forma rápida y fiable.
- Detenga la aplicación: Detenga la aplicación en la base de datos de origen y póngala en modo de solo lectura para que no se puedan realizar más escrituras. Si la base de datos de destino todavía no ha incorporado completamente los cambios efectuados en la base de datos de origen, espere un momento hasta que se terminen de propagar.
- Ejecute las pruebas anteriores a la puesta en servicio: Ejecute las pruebas de migración anteriores a la puesta en servicio para asegurarse de que la migración de datos se ha realizado correctamente.

Puesta en servicio

- **Efectúe la puesta en servicio:** Si las pruebas de migración anteriores a la puesta en servicio se han realizado con éxito, ya puede dirigir su aplicación a Amazon Aurora. Ejecute las secuencias de comandos creadas en la fase anterior a la puesta en servicio para cambiar la configuración de la aplicación y dirigirla a la nueva base de datos de Aurora.
- **Inicie la aplicación:** Ahora ya puede iniciar su aplicación. Si puede hacer que los usuarios dejen de acceder a la aplicación mientras se está ejecutando, hágalo hasta que haya terminado las comprobaciones posteriores a la puesta en servicio.

Comprobaciones posteriores a la puesta en servicio

- **Realice las pruebas posteriores a la puesta en servicio:** ejecute tipos de pruebas manuales o automáticas predefinidas para asegurarse de que la aplicación funciona con la nueva base de datos según lo previsto. Es una buena estrategia empezar a probar la funcionalidad de solo lectura en la base de datos antes de llevar a cabo las pruebas de escritura.
- **Habilite el acceso de los usuarios y supervise con atención:** si las pruebas se llevaron a cabo con éxito, puede dar acceso a la aplicación a los usuarios para completar el proceso de migración. Supervise con atención la aplicación y la base de datos en esta fase.

Conclusión

Amazon Aurora es una base de datos de alto rendimiento, con una elevada disponibilidad y pensada para empresas que ha sido diseñada para la nube. La adopción de Amazon Aurora puede dar como resultado un mejor rendimiento y una mayor disponibilidad respecto a otras bases de datos de código abierto y un costo inferior al de la mayoría de las bases de datos comerciales. Este documento propone estrategias para identificar el mejor método para migrar bases de datos a Amazon Aurora, y explica el procedimiento necesario para planificar y llevar a cabo estas migraciones. En particular, AWS Database Migration Service (AWS DMS), así como la herramienta de conversión de esquemas de AWS son las herramientas recomendadas para los casos de migraciones heterogéneas. Estas potentes herramientas pueden reducir enormemente el costo y la complejidad de las migraciones de bases de datos.

Colaboradores

Las siguientes personas y organizaciones han participado en la redacción de este documento:

- Puneet Agarwal, arquitecto de soluciones, Amazon Web Services
- Scott Williams, arquitecto de soluciones, Amazon Web Services

Documentación adicional

Para obtener más ayuda, consulte las siguientes fuentes:

- [Detalles del producto Amazon Aurora](#)
- [Preguntas frecuentes sobre Amazon Aurora](#)
- [Amazon Database Migration Service](#)
- [Preguntas frecuentes sobre AWS Database Migration Service](#)

Notas

¹ <https://aws.amazon.com/rds/aurora/>

² <http://aws.amazon.com/rds/aurora/pricing/>

³ https://do.awsstatic.com/product-marketing/Aurora/Aurora_Export_Import_Best_Practices_v1-3.pdf

⁴ http://docs.aws.amazon.com/pt_br/AmazonRDS/latest/UserGuide/Aurora.Replication.html#Aurora.Overview.Replication.MySQLReplication

⁵ http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Aurora.Migrate.html#USER_ImportAurora.PreImport

⁶ http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateSnapshot.html

⁷ http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CopySnapshot.html

⁸ http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Aurora.Migrate.html#USER_ImportAuroraCluster.Console

⁹ <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Aurora.Connect.html>

- ¹⁰ <https://dev.mysql.com/doc/refman/5.6/en/mysqldump.html>
- ¹¹ <http://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/Welcome.html>
- ¹² http://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_SchemaConversionTool.GettingStarted.html
- ¹³ http://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_SchemaConversionTool.Installing.html
- ¹⁴ http://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_SchemaConversionTool.Installing.html#CHAP_SchemaConversionTool.Installing.JDBCDrivers
- ¹⁵ <https://forums.aws.amazon.com/forum.jspa?forumID=208>
- ¹⁶ <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Aurora.CreateInstance.html>
- ¹⁷ http://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_SchemaConversionTool.BestPractices.html
- ¹⁸ http://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.html
- ¹⁹ <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Aurora.CreateInstance.html>