Descomposición de la base de datos en AWS

AWS Guía prescriptiva

AWS Guía prescriptiva: Descomposición de la base de datos en AWS

Table of Contents

Descomposición de bases de datos en AWS

Philippe Wanner y Saurabh Sharma, Amazon Web Services

Octubre de 2025 (historia del documento)

La modernización de las bases de datos, en particular la descomposición de las bases de datos monolíticas, es un flujo de trabajo fundamental para las organizaciones que desean mejorar la agilidad, la escalabilidad y el rendimiento de sus sistemas de administración de datos. A medida que las empresas crecen y sus necesidades de datos se vuelven más complejas, las bases de datos monolíticas tradicionales suelen tener dificultades para mantener el ritmo. Esto provoca cuellos de botella en el rendimiento, desafíos de mantenimiento y dificultades para adaptarse a los cambiantes requisitos empresariales.

Los siguientes son los desafíos más comunes de las bases de datos monolíticas:

- Desalineación del dominio empresarial: las bases de datos monolíticas a menudo no logran alinear la tecnología con distintos dominios empresariales, lo que puede limitar el crecimiento de la organización.
- Limitaciones de escalabilidad: los sistemas suelen alcanzar los límites de escalabilidad, lo que crea barreras a la expansión empresarial.
- Rigidez arquitectónica: las estructuras estrechamente acopladas dificultan la actualización de componentes específicos sin afectar a todo el sistema.
- Degradación del rendimiento: el aumento de la carga de datos y el aumento de la simultaneidad de los usuarios suelen provocar un deterioro del rendimiento del sistema.

Las siguientes son las ventajas de la descomposición de las bases de datos:

- Agilidad empresarial mejorada: la descomposición permite una adaptación rápida a las cambiantes necesidades empresariales y permite un escalado independiente.
- Rendimiento optimizado: la descomposición le ayuda a crear soluciones de bases de datos especializadas que se adaptan a casos de uso específicos y a escalar cada base de datos de forma independiente.
- Administración de costos mejorada: la descomposición permite una utilización más eficiente de los recursos y reduce los costos operativos.

- Opciones de licencia flexibles: la descomposición crea oportunidades para pasar de costosas licencias patentadas a alternativas de código abierto.
- Fomento de la innovación: la descomposición facilita la adopción de bases de datos diseñadas específicamente para cargas de trabajo específicas.

Destinatarios previstos

Esta guía ayuda a los arquitectos de bases de datos, arquitectos de soluciones en la nube, equipos de desarrollo de aplicaciones y arquitectos empresariales. Está diseñada para ayudarlo a descomponer las bases de datos monolíticas en almacenes de datos alineados con los microservicios, implementar arquitecturas de bases de datos basadas en el dominio, planificar estrategias de migración de bases de datos y escalar las operaciones de las bases de datos para satisfacer las crecientes demandas empresariales. Para comprender los conceptos y las recomendaciones de esta guía, debe estar familiarizado con los principios de las bases de datos relacionales y NoSQL AWS , los servicios de bases de datos gestionadas y los patrones de arquitectura de microservicios. El objetivo de esta guía es ayudar a las organizaciones que se encuentran en las etapas iniciales de un proyecto de descomposición de bases de datos.

Objetivos

Esta guía puede ayudar a su organización a alcanzar los siguientes objetivos:

- Recopile los requisitos para descomponer la arquitectura de destino.
- Desarrolle una metodología sistemática para evaluar el riesgo y comunicarse.
- Cree un plan de descomposición.
- Defina las métricas de éxito, los indicadores clave de rendimiento (KPIs), una estrategia de mitigación y un plan de continuidad empresarial.
- Establezca una mejor elasticidad de la carga de trabajo que le ayude a seguir la demanda empresarial.
- Aprenda a adoptar bases de datos especializadas para casos de uso específicos, lo que permite la innovación.
- Refuerce la seguridad y el gobierno de los datos de su organización.
- Reduzca los costos de la siguiente manera:
  - Tarifas de licencia reducidas

- Reducción de la dependencia de un proveedor
- Mejora del acceso a un apoyo comunitario más amplio y a las innovaciones
- Posibilidad de elegir diferentes tecnologías de bases de datos para diferentes componentes
- Migración gradual, que reduce el riesgo y distribuye los costos a lo largo del tiempo
- Utilización mejorada de los recursos

Desafíos comunes y responsabilidades de administración para la descomposición de bases de datos

La descomposición de las bases de datos es un proceso complejo que requiere una planificación, ejecución y administración cuidadosas. A medida que las organizaciones buscan modernizar su infraestructura de datos, a menudo se enfrentan a una miríada de desafíos que pueden afectar el éxito de sus proyectos. En esta sección se describen los obstáculos más comunes y se presenta un enfoque estructurado para superarlos.

Desafíos comunes

Un proyecto de descomposición de bases de datos se enfrenta a varios desafíos en las dimensiones técnica, empresarial y de personal. Desde el punto de vista técnico, garantizar la coherencia de los datos en los sistemas distribuidos supone un obstáculo importante. También puede tener un impacto potencial en el rendimiento y la estabilidad durante el período de transición, por lo que debe integrarse sin problemas con los sistemas existentes. Los desafíos relacionados con las personas incluyen la curva de aprendizaje asociada al nuevo sistema, la posible resistencia de los empleados al cambio y la disponibilidad de los recursos necesarios. Desde una perspectiva empresarial, el proyecto debe hacer frente a los riesgos de sobrepasar los plazos, las restricciones presupuestarias y la posibilidad de que se produzcan interrupciones en la actividad empresarial durante el proceso de migración.

Definir funciones y responsabilidades

Dados estos complejos desafíos que abarcan dimensiones técnicas, de personal y empresariales, establecer funciones y responsabilidades claras se vuelve fundamental para el éxito del proyecto. Una matriz responsable, responsable, consultada e informada (RACI) proporciona la estructura necesaria para afrontar estos desafíos. Define de forma explícita quién toma las decisiones, quién realiza el trabajo, quién aporta información y quién debe mantenerse informado en cada etapa de la descomposición. Esta claridad ayuda a evitar las demoras causadas por una toma de decisiones ambigua, fomenta la participación adecuada de las partes interesadas y crea responsabilidad en relación con los resultados clave. Sin este marco, los equipos podrían tener que hacer frente a la superposición de responsabilidades, la falta de comunicación y las vías de escalamiento poco claras, lo que podría agravar las complejidades técnicas existentes y los desafíos de la gestión del cambio, al tiempo que aumentaría el riesgo de sobrepasar los plazos y el presupuesto.

El siguiente ejemplo de matriz RACI es un punto de partida que puede ayudarle a aclarar las posibles funciones y responsabilidades de su organización.

Tarea o actividad	Gestor de proyectos	Arquitecto	Desarrollador	Parte interesada
Identifique los resultados y desafíos empresariales	A/R	R	C	–
Defina el alcance e identifique los requisitos	A	R	C	C/I
Identifique las métricas de éxito del proyecto	A	R	C	I
Cree y ejecute el plan de comunicación	A/R	C	C	I
Defina la arquitectura objetivo	I	A/R	C	–
Controle el acceso a la base	I	A/R	R	–
Cree y ejecute el plan de continuidad del negocio	A/R	C	I	–
Analice la cohesión y el acoplamiento	I	A/R	R	I

Mueva la lógica empresarial (como los procedimientos almacenados) de la base de datos a la capa de aplicación	I	A	R	–
Desacople las relaciones de las tablas, conocidas como uniones	I	A	R	–

Definir el alcance y los requisitos para la descomposición de la base de datos

Al definir el alcance e identificar los requisitos de su proyecto de descomposición de bases de datos, debe basarse en las necesidades de su organización. Esto requiere un enfoque sistemático que equilibre la viabilidad técnica con el valor empresarial. Este paso inicial sienta las bases de todo el proceso y le ayuda a garantizar que los objetivos del proyecto se alineen con las metas y capacidades de la organización.

Esta sección contiene los siguientes temas:

Establecer un marco de análisis básico

La definición del alcance comienza con un flujo de trabajo sistemático que guía el análisis a través de cuatro fases interconectadas. Este enfoque integral garantiza que los esfuerzos de descomposición de las bases de datos se basen en un conocimiento profundo de los sistemas existentes y los requisitos operativos. Las siguientes son las fases del marco de análisis básico:

1. Análisis de actores: identifique minuciosamente todos los sistemas y aplicaciones que interactúan con la base de datos. Esto implica mapear tanto a los productores que realizan las operaciones de escritura como a los consumidores que se encargan de las operaciones de lectura, además de documentar sus patrones de acceso, frecuencias y horas de uso máximo. Esta visión centrada en el cliente le ayuda a comprender el impacto de cualquier cambio e identificar las rutas críticas que requieren una atención especial durante la descomposición.
2. Análisis de la actividad: profundiza en las operaciones específicas que realiza cada actor. Cree matrices detalladas de creación, lectura, actualización y eliminación (CRUD) para cada sistema e

identifique a qué tablas acceden y cómo. Este análisis le ayuda a descubrir los límites naturales de la descomposición y destaca las áreas en las que puede simplificar la arquitectura actual.

3. Mapeo de dependencias: documente las dependencias directas e indirectas entre los sistemas, creando visualizaciones claras de los flujos y las relaciones de datos. Esto ayuda a identificar los posibles puntos de ruptura y las áreas en las que se necesita una planificación cuidadosa para ganarse la confianza. El análisis considera tanto las dependencias técnicas, como las tablas compartidas y las claves externas, como las dependencias de los procesos empresariales, como las secuencias de flujo de trabajo y los requisitos de presentación de informes.
4. Requisitos de coherencia: examine las necesidades de coherencia de cada operación con altos estándares. Determine qué operaciones requieren coherencia inmediata, como las transacciones financieras. Otras operaciones pueden funcionar con una coherencia eventual, como las actualizaciones de análisis. Este análisis influye directamente en la elección de los patrones de descomposición y en las decisiones arquitectónicas a lo largo del proyecto.

Definir los límites del sistema para la descomposición de las bases de datos

Los límites del sistema son perímetros lógicos que definen dónde termina un sistema y comienza otro, e incluyen la propiedad de los datos, los patrones de acceso y los puntos de integración. Al definir los límites del sistema, tome decisiones meditadas pero decisivas que equilibren la planificación integral con las necesidades prácticas de implementación. Considere la base de datos como una unidad lógica que puede abarcar varias bases de datos o esquemas físicos. Esta definición de límite cumple los siguientes objetivos fundamentales:

- Identifica todos los actores externos y sus patrones de interacción
- Mapea exhaustivamente las dependencias entrantes y salientes
- Documenta las limitaciones técnicas y operativas
- Delinea claramente el alcance del esfuerzo de descomposición

Considerando los ciclos de lanzamiento

Comprender los ciclos de publicación es crucial para planificar la descomposición de las bases de datos. Revise los tiempos de renovación tanto del sistema de destino como de los sistemas dependientes. Identifique oportunidades para realizar cambios coordinados. Considere la posibilidad de planificar el desmantelamiento de los sistemas conectados, ya que esto podría influir en

su estrategia de descomposición. Tenga en cuenta las ventanas de cambio existentes y las restricciones de implementación para minimizar las interrupciones del negocio. Asegúrese de que su plan de implementación se ajuste a los cronogramas de lanzamiento de todos los sistemas conectados.

Evaluar las limitaciones técnicas para la descomposición de las bases de datos

Antes de proceder a la descomposición de la base de datos, evalúe las principales limitaciones técnicas que configurarán su enfoque de modernización. Examine las capacidades de su conjunto tecnológico actual, incluidas las versiones de las bases de datos, los marcos, los requisitos de rendimiento y los acuerdos de nivel de servicio. Tenga en cuenta los mandatos de seguridad y cumplimiento, especialmente para los sectores regulados. Revise los volúmenes de datos actuales, las proyecciones de crecimiento y las herramientas de migración disponibles para fundamentar sus decisiones de escalamiento. Por último, confirme sus derechos de acceso al código fuente y a las modificaciones del sistema, ya que estas determinarán las estrategias de descomposición viables.

Entender el contexto organizacional

La descomposición exitosa de las bases de datos requiere que comprenda el panorama organizacional más amplio en el que opera el sistema. Mapee las dependencias interdepartamentales y establezca canales de comunicación claros entre los equipos. Evalúe las capacidades técnicas de su equipo e identifique cualquier necesidad de formación o carencia de habilidades que deba abordar. Considera las implicaciones de la gestión del cambio, incluida la forma de gestionar las transiciones y mantener la continuidad empresarial. Evalúe los recursos disponibles y cualquier limitación, como las limitaciones presupuestarias o de personal. Por último, alinee su estrategia de descomposición con las expectativas y prioridades de las partes interesadas para promover el apoyo continuo durante todo el proyecto.

Evaluar el riesgo de descomposición de la base de datos

Una evaluación integral de los riesgos es esencial para que la descomposición de la base de datos tenga éxito. Evalúe cuidadosamente los riesgos, como la integridad de los datos durante la migración, la posible degradación del rendimiento del sistema, los posibles fallos de integración y las vulnerabilidades de seguridad. Estos desafíos técnicos deben equilibrarse con los riesgos empresariales, incluidas las posibles interrupciones operativas, las limitaciones de recursos, los

retrasos en los plazos y las restricciones presupuestarias. Para cada riesgo identificado, desarrolle estrategias de mitigación y planes de contingencia específicos a fin de mantener el impulso del proyecto y, al mismo tiempo, proteger las operaciones comerciales.

Cree una matriz de riesgos que evalúe tanto el impacto como la probabilidad de posibles problemas. Trabaje con los equipos técnicos y las partes interesadas de la empresa para identificar los riesgos, establecer umbrales claros para la intervención y desarrollar estrategias de mitigación específicas. Por ejemplo, califique el riesgo de pérdida de datos como de alto impacto y baja probabilidad, y requiere estrategias de respaldo sólidas. Una degradación menor del rendimiento puede tener un impacto medio y una probabilidad alta, y requiere una supervisión proactiva.

Establezca ciclos periódicos de revisión de riesgos para reevaluar las prioridades y ajustar los planes de mitigación a medida que el proyecto evoluciona. Este enfoque sistemático garantiza que los recursos se centren en los riesgos más críticos y, al mismo tiempo, mantiene vías de escalamiento claras para los problemas emergentes.

Definir los criterios de éxito para la descomposición de las bases de datos

Los criterios de éxito para la descomposición de la base de datos deben estar claramente definidos y medirse en múltiples dimensiones. Desde una perspectiva empresarial, establezca objetivos específicos para la reducción de costos, la mejora time-to-market, la disponibilidad del sistema y la satisfacción del cliente. El éxito técnico debe medirse mediante mejoras cuantificables en el rendimiento del sistema, la eficiencia de la implementación, la coherencia de los datos y la confiabilidad general. Para el proceso de migración, defina requisitos estrictos para evitar la pérdida de datos, establecer límites aceptables para las interrupciones de la actividad empresarial, cumplir con el presupuesto y respetar los plazos.

Documente estos criterios exhaustivamente manteniendo las métricas de referencia y objetivo, las metodologías de medición claras y los cronogramas de revisión periódicos. Asigne propietarios claros para cada métrica de éxito y mapee las dependencias entre las diferentes métricas. Este enfoque integral para medir el éxito alinea los logros técnicos con los resultados empresariales y, al mismo tiempo, mantiene la responsabilidad durante todo el proceso de descomposición.

Control del acceso a la base de datos durante la descomposición

Muchas organizaciones se enfrentan a una situación común: una base de datos central que ha crecido de forma orgánica a lo largo de muchos años y a la que acceden directamente varios servicios y equipos. Esto crea varios problemas críticos:

- Crecimiento descontrolado: a medida que los equipos añaden nuevas funciones y modifican los esquemas de forma continua, la base de datos se vuelve cada vez más compleja y difícil de gestionar.
- Problemas de rendimiento: incluso con las mejoras de hardware, la carga creciente amenaza con superar las capacidades de la base de datos. Imposibilidad de ajustar las consultas debido a la complejidad del esquema o a la falta de habilidades. No se puede predecir ni explicar el rendimiento del sistema.
- Parálisis de la descomposición: resulta prácticamente imposible dividir o refactorizar la base de datos mientras varios equipos la están modificando activamente.

> **Note**
>
> Los sistemas de bases de datos monolíticas suelen reutilizar las mismas credenciales para las aplicaciones o los servicios o para la administración. Esto conduce a una mala trazabilidad de las bases de datos. Establecer funciones específicas y adoptar el principio del privilegio mínimo puede ayudarle a aumentar la seguridad y la disponibilidad.

Cuando se trata de una base de datos monolítica que se ha vuelto difícil de manejar, uno de los patrones más eficaces para controlar el acceso es el denominado servicio de empaquetado de bases de datos. Proporciona un primer paso estratégico en la gestión de sistemas de bases de datos complejos. Establece un acceso controlado a las bases de datos y permite una modernización gradual, a la vez que reduce el riesgo. Este enfoque sienta las bases para las mejoras incrementales al proporcionar una visibilidad clara de las dependencias y los patrones de uso de los datos. Se trata de una arquitectura de transición que sirve como un paso hacia la descomposición total de la base de datos. El servicio de empaquetado proporciona la estabilidad y el control necesarios para que ese viaje sea exitoso.

Esta sección contiene los siguientes temas:

Controlar el acceso con el patrón de servicio de empaquetado de bases de datos

Un servicio contenedor es una capa de servicio que actúa como fachada de la base de datos. Este enfoque es particularmente valioso cuando se necesita mantener la funcionalidad existente y, al mismo tiempo, prepararse para una futura descomposición. Este patrón sigue un principio simple: cuando algo está demasiado desordenado, comience por contener el desorden. El servicio de empaquetado se convierte en la única forma autorizada de acceder a la base de datos, ya que proporciona una interfaz controlada y oculta la complejidad subyacente.

Utilice este patrón cuando la descomposición inmediata de la base de datos no sea factible debido a esquemas complejos o cuando varios servicios requieran un acceso continuo a los datos. Es especialmente valioso durante los períodos de transición, ya que proporciona tiempo para una refactorización cuidadosa y, al mismo tiempo, mantiene la estabilidad del sistema. Este patrón funciona bien cuando se consolida la propiedad de los datos en manos de equipos específicos o cuando las nuevas aplicaciones necesitan vistas agregadas en varias tablas.

Por ejemplo, aplique este patrón cuando:

- La complejidad del esquema impide la separación inmediata
- Varios equipos necesitan un acceso continuo a los datos
- Se prefiere la modernización gradual
- La reestructuración del equipo requiere una propiedad clara de los datos
- Las nuevas aplicaciones necesitan vistas de datos consolidadas

Ventajas y limitaciones del patrón de servicio de empaquetado de bases de datos

Las siguientes son las ventajas del patrón de envoltura de bases de datos:

- Crecimiento controlado: el servicio de contenedor evita nuevas adiciones incontroladas al esquema de la base de datos.
- Límites claros: el proceso de implementación le ayuda a establecer límites claros de propiedad y responsabilidad.
- Libertad de refactorización: un servicio de embalaje le permite realizar cambios internos sin afectar a los consumidores.
- Observabilidad mejorada: un servicio de empaquetado es un punto único de monitoreo y registro.
- Pruebas simplificadas: un servicio de empaquetado facilita a los servicios consumidores la creación de versiones simuladas y simplificadas para las pruebas.

Las siguientes son las limitaciones del patrón de empaquetado de la base de datos.

- Acoplamiento tecnológico: un servicio de empaquetado funciona mejor cuando utiliza el mismo conjunto de tecnologías que los servicios consumidores.
- Gastos generales iniciales: el servicio de empaquetado requiere una infraestructura adicional que podría afectar al rendimiento.
- Esfuerzo de migración: para implementar el servicio de empaquetado, es necesario coordinar los equipos para evitar el acceso directo.
- Rendimiento: si el servicio de empaquetado tiene mucho tráfico, un uso intensivo o un acceso frecuente, es posible que los servicios que consumen productos tengan un rendimiento deficiente. Además de la base de datos, el servicio de empaquetado debe gestionar la paginación, los cursores y las conexiones a la base de datos. Según el caso de uso, es posible que no se escale bien y que no sea adecuado para cargas de trabajo de extracción, transformación y carga (ETL).

Implementación del patrón de servicio de empaquetado de bases de datos

Hay dos fases para implementar el patrón de servicio de empaquetado de bases de datos. En primer lugar, se crea el servicio de empaquetado de bases de datos. Luego, dirige todos los accesos a través de él y documenta los patrones de acceso.

Fase 1: Creación del servicio de empaquetado de bases de datos

Cree una capa de servicio ligera que actúe como guardián de su base de datos. Inicialmente, debería reflejar todas las funcionalidades existentes. Este servicio contenedor se convierte en el punto de acceso obligatorio para todas las operaciones de la base de datos, lo que convierte las dependencias directas de la base de datos en dependencias de nivel de servicio. Implemente un

registro y una supervisión detallados en esta capa para realizar un seguimiento de los patrones de uso, las métricas de rendimiento y las frecuencias de acceso. Mantenga sus procedimientos almacenados actuales, pero asegúrese de que solo se pueda acceder a ellos a través de esta nueva interfaz de servicio.

Fase 2: Implementación del control de acceso

Redirija sistemáticamente todo el acceso a la base de datos a través del servicio de empaquetado y, a continuación, revoque los permisos directos a la base de datos de los sistemas externos que accedan directamente a la base de datos. Documente cada patrón de acceso y cada dependencia a medida que se migran los servicios. Este acceso controlado permite la refactorización interna de los componentes de la base de datos sin interrumpir a los consumidores externos. Por ejemplo, comience con operaciones de solo lectura y bajo riesgo en lugar de flujos de trabajo transaccionales complejos.

Fase 3: Supervise el rendimiento de la base de datos

Utilice el servicio de encapsulado como punto de supervisión centralizado del rendimiento de la base de datos. Realice un seguimiento de las métricas clave, incluidos los tiempos de respuesta a las consultas, los patrones de uso, las tasas de error y la utilización de los recursos. Configure alertas para los umbrales de rendimiento y los patrones inusuales. Por ejemplo, supervise las consultas de ejecución lenta, la utilización del grupo de conexiones y el rendimiento de las transacciones para identificar de forma proactiva los posibles problemas.

Utilice esta vista consolidada para optimizar el rendimiento de la base de datos mediante el ajuste de las consultas, los ajustes de asignación de recursos y el análisis de los patrones de uso. La naturaleza centralizada del servicio de empaquetado facilita la implementación de mejoras y la validación de su impacto en todos los consumidores, al tiempo que mantiene estándares de rendimiento consistentes.

Mejores prácticas para implementar un servicio de empaquetado de bases de datos

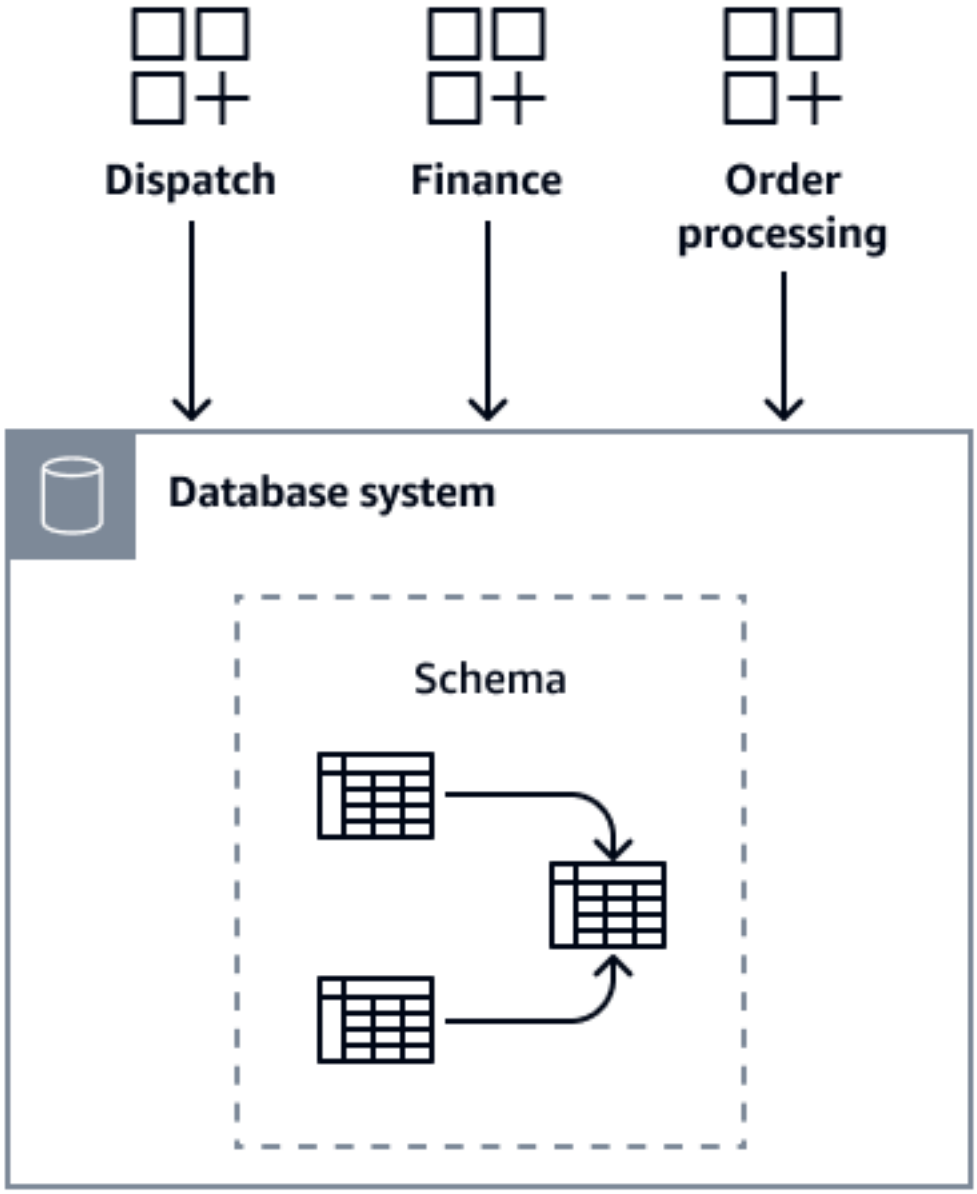
Las siguientes prácticas recomendadas pueden ayudarle a implementar un servicio de empaquetado de bases de datos:

- Comience de a poco: comience con un contenedor mínimo que simplemente represente la funcionalidad existente
- Mantenga la estabilidad: mantenga estable la interfaz de servicio y, al mismo tiempo, realice mejoras internas

- Supervise el uso: implemente una supervisión integral para comprender los patrones de acceso
- Propiedad clara: asigne un equipo dedicado al mantenimiento tanto del contenedor como del esquema subyacente
- Fomente el almacenamiento local: motive a los equipos a almacenar sus datos en sus propias bases de datos

Ejemplo basado en escenarios

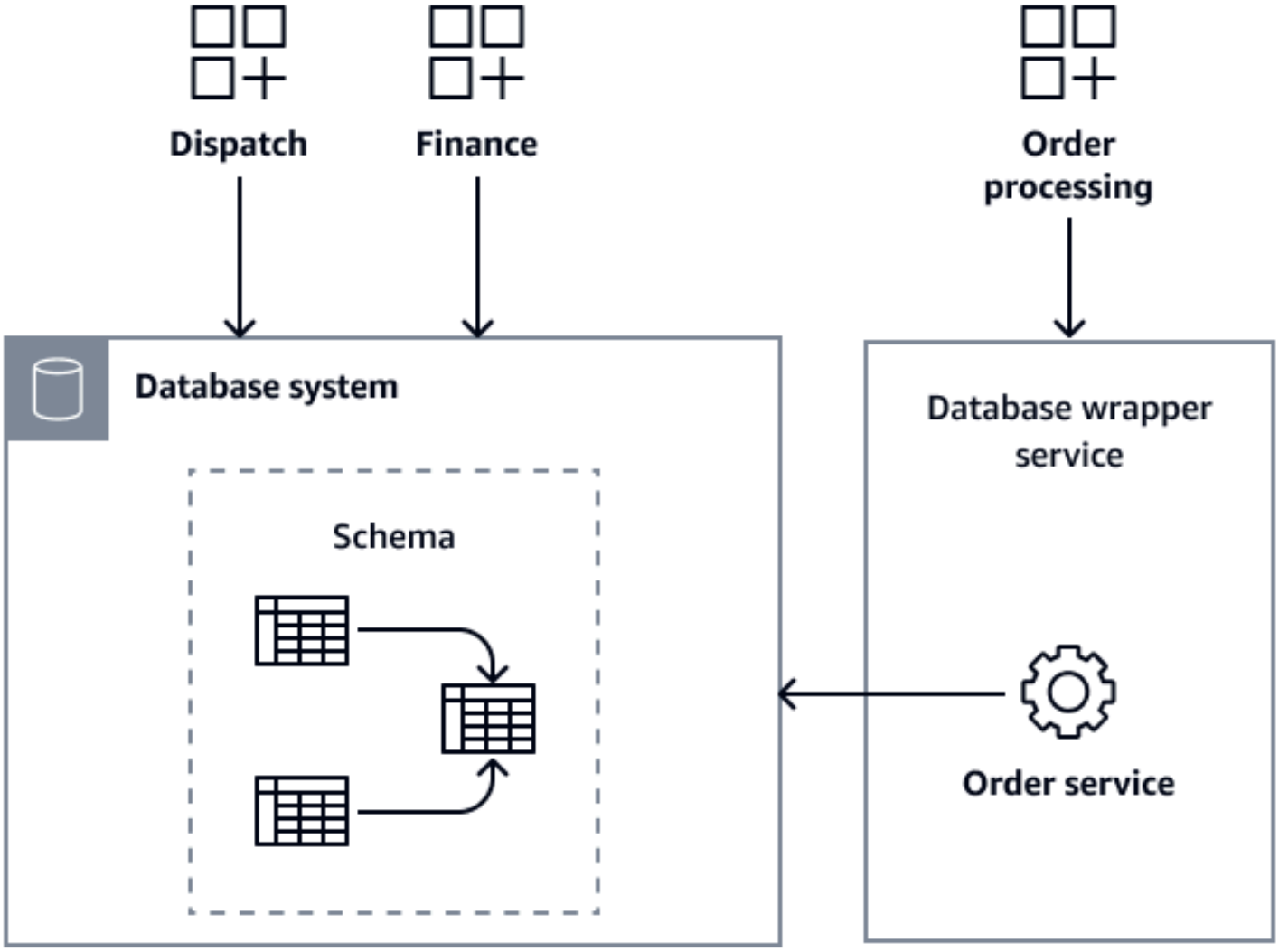
En esta sección se describe un ejemplo de cómo una empresa ficticia, llamada AnyCompany Books, podría utilizar el patrón de envoltura de la base de datos para controlar el acceso a su monolítico sistema de bases de datos. En AnyCompany Books, hay tres servicios fundamentales: despacho, financiación y procesamiento de pedidos. Estos servicios comparten el acceso a una base de datos central. Cada servicio es mantenido por un equipo diferente. Con el tiempo, modifican de forma independiente el esquema de la base de datos para satisfacer sus necesidades específicas. Esto ha dado lugar a una intrincada red de dependencias y a una estructura de base de datos cada vez más compleja.



El arquitecto empresarial o de aplicaciones de la empresa reconoce la necesidad de descomponer esta base de datos monolítica. Su objetivo es dotar a cada servicio de su propia base de datos dedicada para mejorar la capacidad de mantenimiento y reducir las dependencias entre equipos. Sin embargo, se enfrentan a un desafío importante: es casi imposible descomponer la base de datos mientras los tres equipos siguen modificándola activamente para sus proyectos en curso. Los constantes cambios de esquema y la falta de coordinación entre los equipos hacen que sea extremadamente arriesgado intentar una reestructuración importante.

El arquitecto utiliza el patrón de servicio de empaquetado de bases de datos para empezar a controlar el acceso a la base de datos monolítica. En primer lugar, configuraron el servicio de

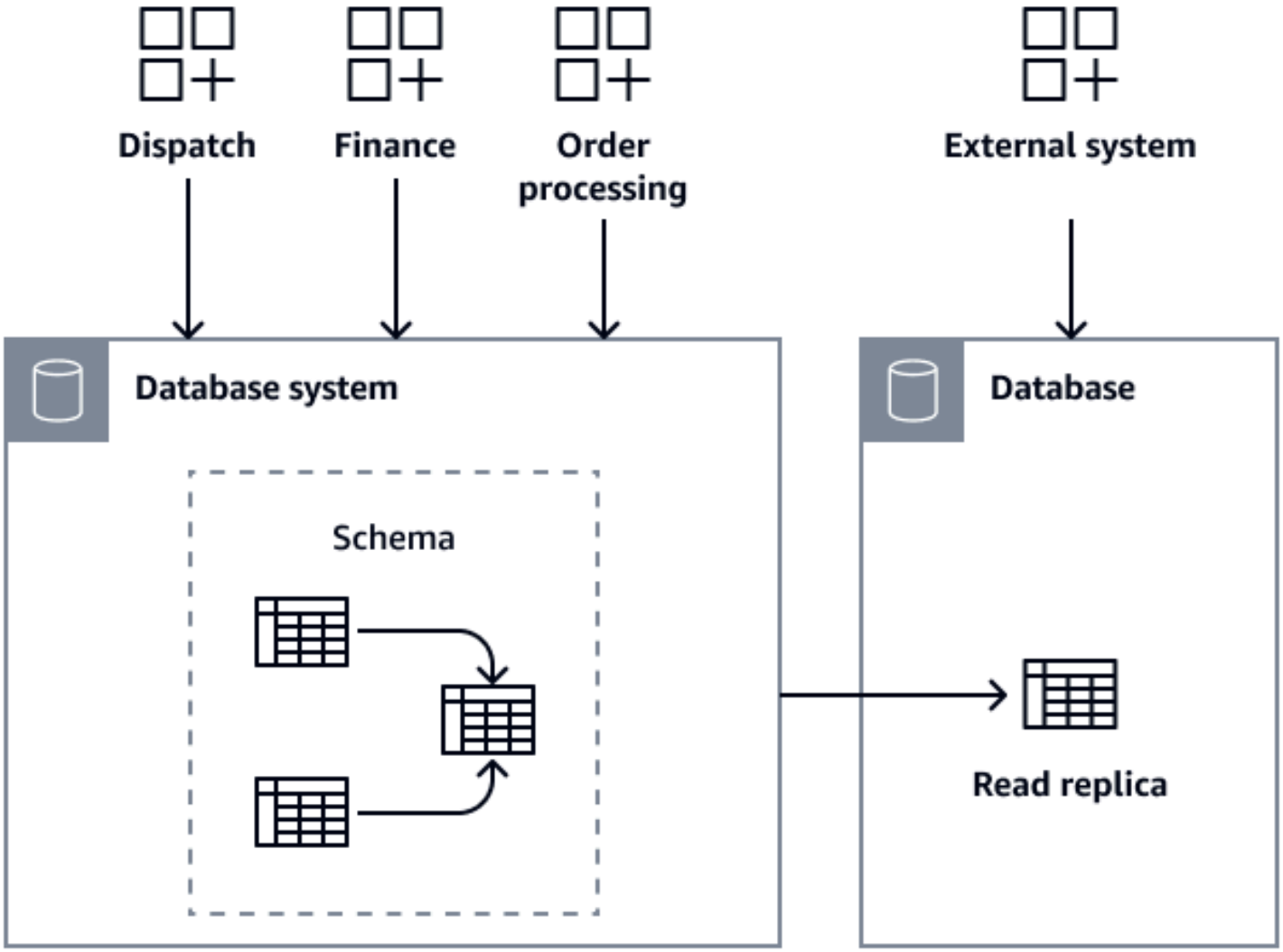
empaquetado de bases de datos para un módulo en particular, denominado servicio de pedidos. A continuación, redirigen el servicio de procesamiento de pedidos para acceder al servicio empaquetador en lugar de acceder directamente a la base de datos. La siguiente imagen muestra la infraestructura modificada.



Controlar el acceso con el patrón CQRS

Otro patrón que puede utilizar para aislar los sistemas externos que se conectan a esta base de datos central es la segregación de responsabilidades por consultas de comandos (CQRS). Si algunos de los sistemas externos se conectan a la base de datos central principalmente para realizar lecturas, como análisis, informes u otras operaciones de lectura intensiva, puede crear almacenes de datos independientes optimizados para la lectura.

Este patrón aísla eficazmente estos sistemas externos de los impactos de la descomposición de la base de datos y los cambios en el esquema. Al mantener réplicas de lectura específicas o almacenes de datos diseñados específicamente para patrones de consulta específicos, los equipos pueden continuar sus operaciones sin verse afectados por los cambios en la estructura de la base de datos principal. Por ejemplo, mientras se descompone la base de datos monolítica, los sistemas de generación de informes pueden seguir trabajando con las vistas de datos existentes y las cargas de trabajo analíticas pueden mantener sus patrones de consulta actuales a través de almacenes analíticos específicos. Este enfoque proporciona aislamiento técnico y permite la autonomía de la organización, ya que los diferentes equipos pueden desarrollar sus sistemas de forma independiente sin tener que estar estrechamente relacionados con el proceso de transformación de la base de datos principal.



Para obtener más información sobre este patrón y un ejemplo de su uso para desacoplar las relaciones de las tablas, consulte Patrón CQRS más adelante en esta guía.

Análisis de la cohesión y el acoplamiento para la descomposición de la base de datos

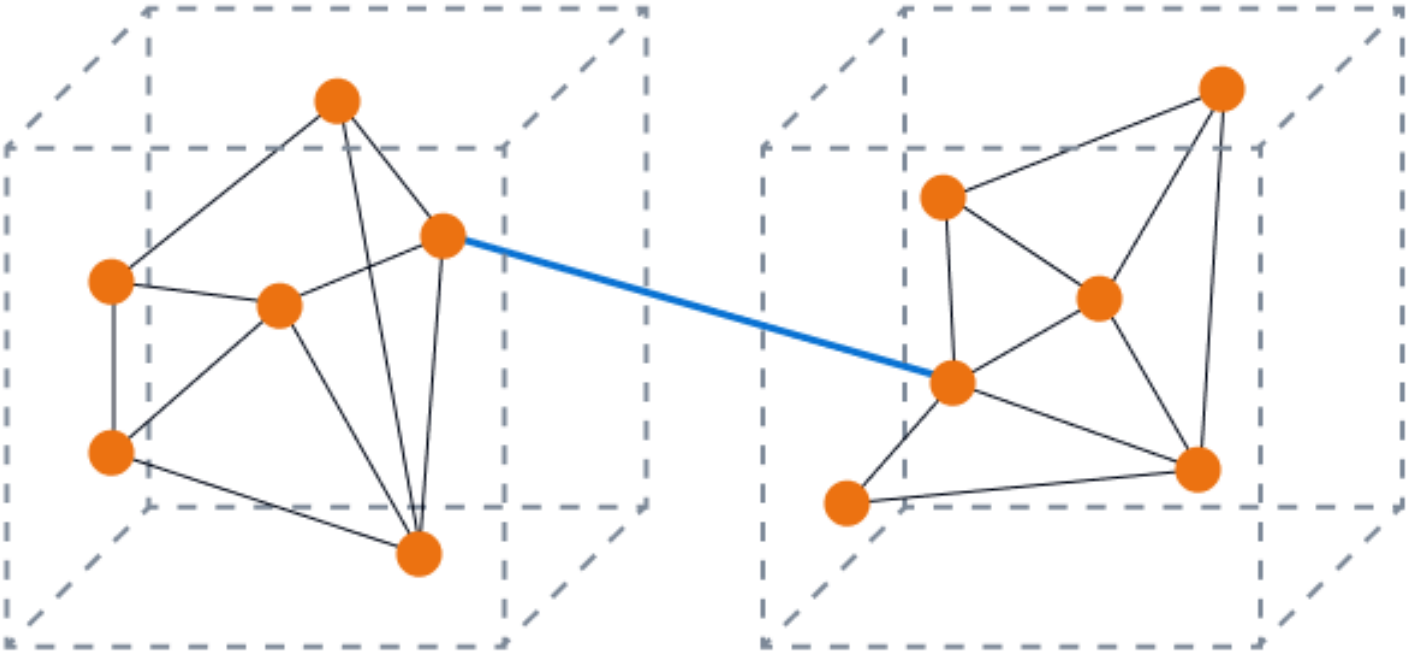
Esta sección le ayuda a analizar los patrones de acoplamiento y cohesión de su base de datos monolítica para guiar su descomposición. Comprender cómo los componentes de la base de datos interactúan y dependen unos de otros es crucial para identificar los puntos de ruptura naturales, evaluar la complejidad y planificar un enfoque de migración gradual. Este análisis revela las dependencias ocultas, resalta las áreas que son adecuadas para una separación inmediata y le ayuda a priorizar las iniciativas de descomposición y, al mismo tiempo, a minimizar los riesgos de transformación. Al examinar tanto el acoplamiento como la cohesión, puede tomar decisiones informadas sobre la secuencia de separación de los componentes a fin de mantener la estabilidad del sistema durante todo el proceso de transformación.

Esta sección contiene los siguientes temas:

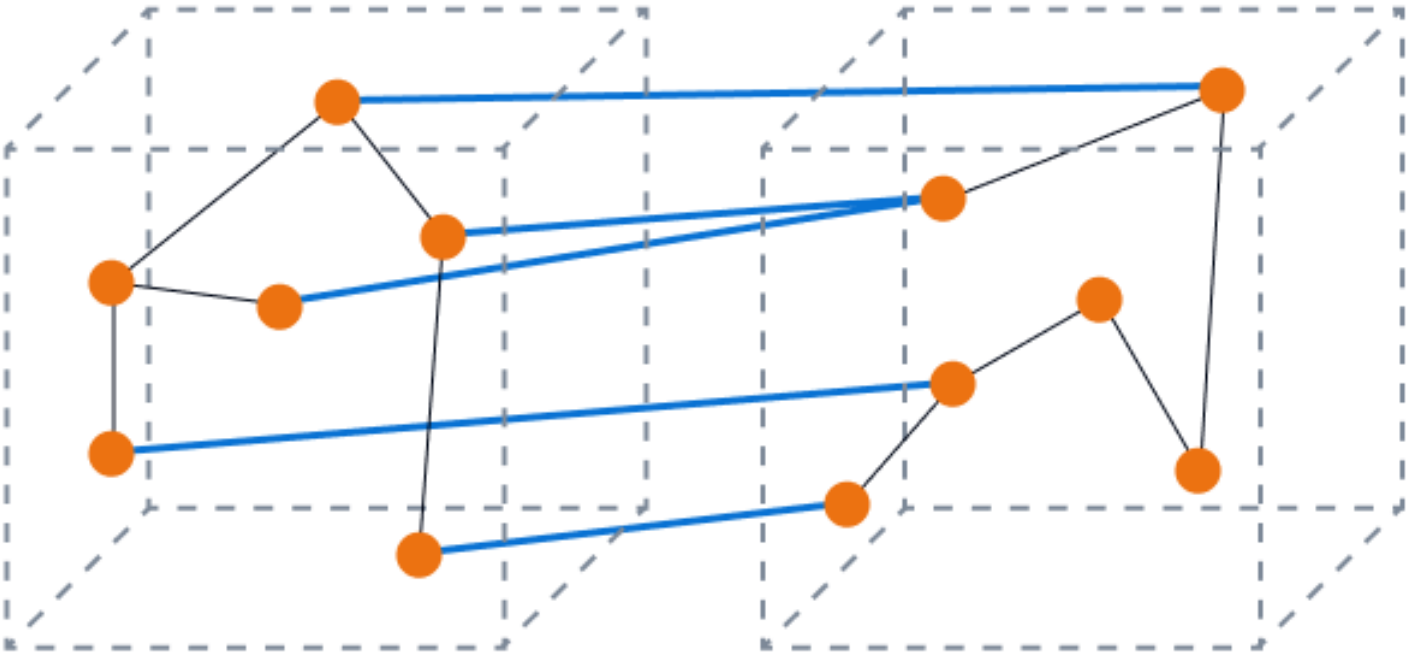
Acerca de la cohesión y el acoplamiento

El acoplamiento mide el grado de interdependencia entre los componentes de la base de datos. En un sistema bien diseñado, se desea lograr un acoplamiento flexible, en el que los cambios en un componente tengan un impacto mínimo en los demás. La cohesión mide qué tan bien funcionan juntos los elementos de un componente de base de datos para cumplir un propósito único y bien definido. Una alta cohesión indica que los elementos de un componente están estrechamente relacionados y se centran en una función específica. Al descomponer una base de datos monolítica, debe analizar tanto la cohesión dentro de los componentes individuales como el acoplamiento entre ellos. Este análisis le ayuda a tomar decisiones informadas sobre cómo desglosar la base de datos y, al mismo tiempo, mantener la integridad y el rendimiento del sistema.

La siguiente imagen muestra un acoplamiento suelto con una alta cohesión. Los componentes de la base de datos trabajan juntos para realizar una función específica y se minimiza el impacto del cambio en un solo componente. Este es el estado ideal.



La siguiente imagen muestra un alto acoplamiento con baja cohesión. Los componentes de la base de datos están desconectados y es muy probable que los cambios afecten a otros componentes.



Patrones de acoplamiento comunes en bases de datos monolíticas

Existen varios patrones de acoplamiento que se encuentran comúnmente al descomponer una base de datos monolítica en bases de datos específicas para microservicios. Comprender estos patrones

es crucial para el éxito de las iniciativas de modernización de bases de datos. En esta sección se describe cada patrón, sus desafíos y las mejores prácticas para reducir el acoplamiento.

Implementación: patrón de acoplamiento

Definición: Los componentes están estrechamente interconectados a nivel de código y esquema. Por ejemplo, la modificación de la estructura de una `customer` tabla afecta a `orderinventory`, y a `billing` los servicios.

Impacto de la modernización: cada microservicio requiere su propio esquema de base de datos y capa de acceso a los datos dedicados.

Desafíos:

- Los cambios en las tablas compartidas afectan a varios servicios
- Alto riesgo de efectos secundarios no deseados
- Mayor complejidad de las pruebas
- Es difícil modificar los componentes individuales

Mejores prácticas para reducir el acoplamiento:

- Defina interfaces claras entre los componentes
- Utilice capas de abstracción para ocultar los detalles de la implementación
- Implemente esquemas específicos de dominio

Patrón de acoplamiento temporal

Definición: Las operaciones deben ejecutarse en una secuencia específica. Por ejemplo, el procesamiento de pedidos no puede continuar hasta que se hayan completado las actualizaciones de inventario.

Impacto de la modernización: cada microservicio necesita un control de datos autónomo.

Desafíos:

- Romper las dependencias sincrónicas entre los servicios
- Cuellos de botella en el rendimiento

- Difícil de optimizar
- Procesamiento paralelo limitado

Mejores prácticas para reducir el acoplamiento:

- Implemente el procesamiento asíncrono siempre que sea posible
- Utilice arquitecturas basadas en eventos
- Diseñe para lograr una coherencia eventual cuando sea apropiado

Patrón de acoplamiento de despliegue

Definición: Los componentes del sistema se deben implementar como una sola unidad. Por ejemplo, un cambio menor en la lógica de procesamiento de pagos requiere la redistribución de toda la base de datos.

Impacto de la modernización: despliegues de bases de datos independientes por servicio

Desafíos:

- Implementaciones de alto riesgo
- Frecuencia de despliegue limitada
- Procedimientos de reversión complejos

Mejores prácticas para reducir el acoplamiento:

- Divídalos en componentes que se pueden implementar de forma independiente
- Implemente estrategias de fragmentación de bases de datos
- Utilice patrones de despliegue azul-verde

Patrón de acoplamiento de dominios

Definición: Los dominios empresariales comparten estructuras y lógica de bases de datos. Por ejemplo, los `inventory` dominios `customerorder`, y comparten tablas y procedimientos almacenados.

Impacto de la modernización: aislamiento de datos de dominios específicos

Desafíos:

- Límites de dominio complejos
- Es difícil escalar dominios individuales
- Reglas comerciales complicadas

Mejores prácticas para reducir el acoplamiento:

- Identifique límites de dominio claros
- Separe los datos por contexto de dominio
- Implemente servicios específicos de dominio

Patrones de cohesión comunes en bases de datos monolíticas

Hay varios patrones de cohesión que se encuentran comúnmente al evaluar los componentes de la base de datos para su descomposición. Comprender estos patrones es crucial para identificar los componentes de la base de datos bien estructurados. En esta sección se describe cada patrón, sus características y las mejores prácticas para fortalecer la cohesión.

Patrón de cohesión funcional

Definición: Todos los elementos apoyan y contribuyen directamente a la realización de una función única y bien definida. Por ejemplo, todos los procedimientos y tablas almacenados en un módulo de procesamiento de pagos solo gestionan las operaciones relacionadas con los pagos.

Impacto de la modernización: patrón ideal para el diseño de bases de datos de microservicios

Desafíos:

- Identificar límites funcionales claros
- Separar los componentes de uso mixto
- Mantener una responsabilidad única

Mejores prácticas para reforzar la cohesión:

- Agrupe las funciones relacionadas
- Elimine la funcionalidad no relacionada

- Defina límites claros de los componentes

Patrón de cohesión secuencial

Definición: La salida de un elemento se convierte en entrada para otro. Por ejemplo, los resultados de la validación de un pedido se incorporan al procesamiento del pedido.

Impacto de la modernización: requiere un análisis cuidadoso del flujo de trabajo y un mapeo del flujo de datos

Desafíos:

- Gestionar las dependencias entre los pasos
- Manejo de escenarios de falla
- Mantener el orden del proceso

Mejores prácticas para reforzar la cohesión:

- Documente flujos de datos claros
- Implemente un manejo de errores adecuado
- Diseñe interfaces claras entre los pasos

Patrón de cohesión comunicacional

Definición: Los elementos funcionan con los mismos datos. Por ejemplo, todas las funciones de gestión del perfil de los clientes funcionan con los datos de los clientes.

Impacto de la modernización: ayuda a identificar los límites de los datos para separar los servicios a fin de reducir el acoplamiento entre los módulos

Desafíos:

- Determinar la propiedad de los datos
- Administrar el acceso a los datos compartidos
- Mantener la coherencia de los datos

Mejores prácticas para reforzar la cohesión:

- Defina una propiedad clara de los datos
- Implemente patrones de acceso a los datos adecuados
- Diseñe un particionamiento de datos efectivo

Patrón de cohesión procedimental

Definición: Los elementos se agrupan porque deben ejecutarse en un orden específico, pero es posible que no estén relacionados funcionalmente. Por ejemplo, en el procesamiento de pedidos, un procedimiento almacenado que gestiona tanto la validación de los pedidos como la notificación al usuario se agrupa simplemente porque se producen de forma secuencial, aunque tienen diferentes propósitos y podrían ser gestionados por servicios independientes.

Impacto de la modernización: requiere una separación cuidadosa de los procedimientos y, al mismo tiempo, mantener el flujo del proceso

Desafíos:

- Mantener el flujo correcto del proceso después de la descomposición
- Identificar los verdaderos límites funcionales en comparación con las dependencias procedimentales

Mejores prácticas para reforzar la cohesión:

- Separe los procedimientos en función de su propósito funcional y no de la orden de ejecución
- Utilice patrones de orquestación para gestionar el flujo del proceso
- Implemente sistemas de gestión del flujo de trabajo para secuencias complejas
- Diseñe arquitecturas basadas en eventos para gestionar los pasos del proceso de forma independiente

Patrón de cohesión temporal

Definición: Los elementos están relacionados por requisitos de tiempo. Por ejemplo, cuando se realiza un pedido, se deben ejecutar varias operaciones al mismo tiempo: la comprobación del inventario, el procesamiento de pagos, la confirmación del pedido y la notificación de envío deben realizarse dentro de un período de tiempo específico para mantener un estado de pedido uniforme.

Impacto en la modernización: podría requerir un manejo especial en los sistemas distribuidos

Desafíos:

- Coordinar las dependencias temporales en los servicios distribuidos
- Gestión de transacciones distribuidas
- Confirmar la finalización del proceso en varios componentes

Mejores prácticas para reforzar la cohesión:

- Implemente los mecanismos de programación y los tiempos de espera adecuados
- Utilice arquitecturas basadas en eventos con un manejo de secuencias claro
- Diseñe para lograr una coherencia final con los patrones de compensación
- Implemente patrones tipo saga para transacciones distribuidas

Patrón de cohesión lógico o coincidente

Definición: Los elementos se clasifican lógicamente para hacer las mismas cosas, aunque tengan relaciones débiles o nulas. Un ejemplo es almacenar los datos de los pedidos de los clientes, los recuentos del inventario del almacén y las plantillas de correo electrónico de marketing en el mismo esquema de base de datos, ya que todos se relacionan con las operaciones de ventas, a pesar de tener diferentes patrones de acceso, administración del ciclo de vida y requisitos de escalado diferentes. Otro ejemplo es combinar el procesamiento de pagos de pedidos y la gestión del catálogo de productos en el mismo componente de base de datos, ya que ambos forman parte del sistema de comercio electrónico, aunque desempeñan funciones empresariales distintas con necesidades operativas diferentes.

Impacto de la modernización: debería refactorizarse o reorganizarse

Desafíos:

- Identificar mejores patrones de organización
- Romper dependencias innecesarias
- Reestructurar componentes que se agruparon arbitrariamente

Mejores prácticas para reforzar la cohesión:

- Reorganícese en función de los verdaderos límites funcionales y dominios empresariales
- Elimine las agrupaciones arbitrarias basadas en relaciones superficiales
- Implemente una separación adecuada de los elementos en función de las capacidades empresariales
- Alinee los componentes de la base de datos con sus requisitos operativos específicos

Implementación de bajo acoplamiento y alta cohesión

Prácticas recomendadas

Las siguientes prácticas recomendadas pueden ayudarle a lograr un bajo nivel de acoplamiento:

- Minimice las dependencias entre los componentes de la base de datos
- Utilice interfaces bien definidas para la interacción de los componentes
- Evite las estructuras de datos globales y estatales compartidas

Las siguientes prácticas recomendadas pueden ayudarle a lograr una alta cohesión:

- Agrupe los datos y las operaciones relacionados
- Asegúrese de que cada componente tenga una responsabilidad única y clara
- Mantenga límites claros entre los diferentes dominios empresariales

Fase 1: mapear las dependencias de los datos

Mapee las relaciones de los datos e identifique los límites naturales. Puede usar herramientas, por ejemplo SchemaSpy, para visualizar la base de datos mostrando las tablas en un diagrama entidad-relación (ER). Esto proporciona un análisis estático de la base de datos e indica algunos de los límites y dependencias claros dentro de la base de datos.

También puede exportar los esquemas de la base de datos a una base de datos de gráficos o a un Jupiter bloc de notas. A continuación, puede aplicar algoritmos de agrupamiento o de componentes interconectados para identificar las dependencias y los límites naturales. Otras AWS Partner herramientas, como las siguientes CAST Imaging, pueden ayudar a comprender las dependencias de la base de datos.

Fase 2: Analice los límites de las transacciones y los patrones de acceso

Analice los patrones de transacciones para mantener las propiedades de atomicidad, coherencia, aislamiento y durabilidad (ACID) y comprenda cómo se accede a los datos y cómo se modifican. Puede utilizar herramientas de análisis y diagnóstico de bases de datos, como Oracle Automatic Workload Repository (AWR)o. PostgreSQL pg_stat_statements Este análisis le ayuda a comprender quién accede a la base de datos y cuáles son los límites de las transacciones. También puede ayudarle a comprender la cohesión y el acoplamiento entre las tablas en tiempo de ejecución. También puede utilizar herramientas de supervisión y creación de perfiles que pueden vincular los perfiles de ejecución de código y base de datos, por ejemplo. Dynatrace AppEngine

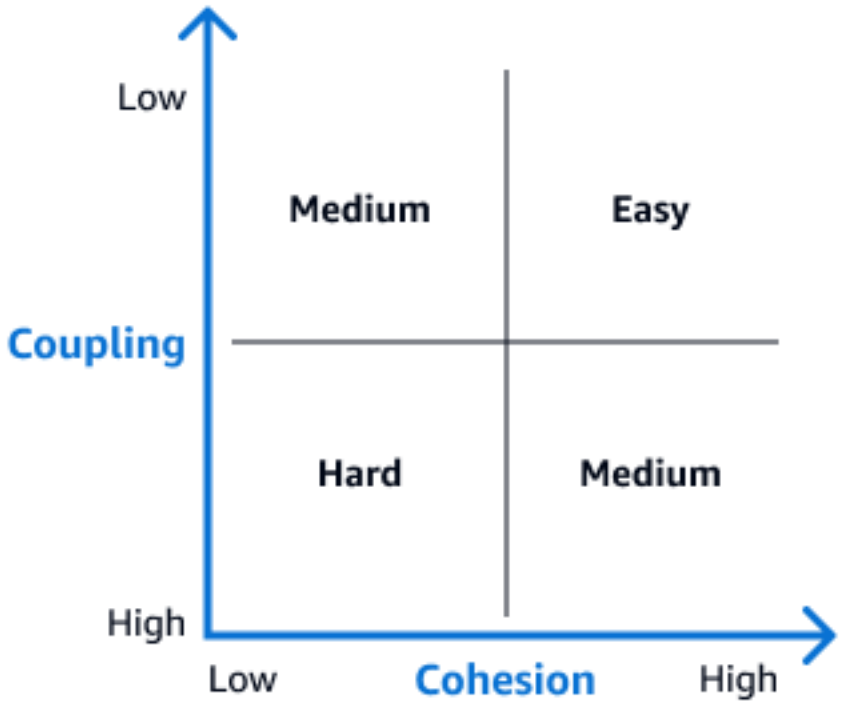
Las herramientas de IA, por ejemplo vFunction, pueden ayudarle a identificar los límites de dominio mediante el análisis de los límites funcionales y de dominio de la aplicación. Si bien analiza vFunction principalmente la capa de aplicación, sus conocimientos pueden guiar la descomposición tanto de la aplicación como de la base de datos, lo que permite alinearla con los dominios empresariales.

Fase 3: Identificar tablas independientes

Busque tablas que demuestren dos características clave:

- Alta cohesión: los contenidos de la tabla están estrechamente relacionados entre sí
- Acoplamiento bajo: tienen una dependencia mínima con respecto a otras mesas.

La siguiente matriz de acoplamiento-cohesión puede ayudarle a identificar la dificultad de desacoplar cada tabla. Las tablas que aparecen en el cuadrante superior derecho de esta matriz son candidatas ideales para los esfuerzos iniciales de desacoplamiento porque son las más fáciles de separar. En un diagrama ER, estas tablas tienen pocas relaciones de clave externa u otras dependencias. Una vez desacopladas estas tablas, avance hacia tablas con relaciones más complejas.



> **Note**
>
> La estructura de la base de datos suele reflejar la arquitectura de la aplicación. Las tablas que son más fáciles de desacoplar a nivel de base de datos suelen corresponder a componentes que son más fáciles de convertir en microservicios a nivel de aplicación.

Migración de la lógica empresarial de la base de datos a la capa de aplicación

La migración de la lógica empresarial de los procedimientos, desencadenantes y funciones almacenados en la base de datos a los servicios de la capa de aplicación es un paso fundamental para descomponer las bases de datos monolíticas. Esta transformación mejora la autonomía del servicio, simplifica el mantenimiento y mejora la escalabilidad. En esta sección se proporciona orientación sobre el análisis de la lógica de la base de datos, la planificación de la estrategia de migración y, posteriormente, la implementación de la transformación sin dejar de mantener la continuidad empresarial. También se analiza el establecimiento de un plan de reversión eficaz.

Esta sección contiene los siguientes temas:

Fase 1: Análisis de la lógica empresarial

Al modernizar las bases de datos monolíticas, primero debe realizar un análisis exhaustivo de la lógica de la base de datos existente. Esta fase se centra en tres categorías principales:

- Los procedimientos almacenados suelen contener operaciones empresariales críticas, como la lógica de manipulación de datos, las reglas empresariales, las comprobaciones de validación y los cálculos. Como componentes fundamentales de la lógica empresarial de la aplicación, requieren una descomposición cuidadosa. Por ejemplo, los procedimientos almacenados de una organización financiera pueden gestionar los cálculos de intereses, la conciliación de cuentas y las comprobaciones de conformidad.
- Los desencadenantes son componentes clave de la base de datos que gestionan los registros de auditoría, la validación de los datos, los cálculos y la coherencia entre tablas. Por ejemplo, una organización minorista puede utilizar activadores para gestionar las actualizaciones de inventario en todo su sistema de procesamiento de pedidos, lo que demuestra la complejidad de las operaciones automatizadas de las bases de datos.

- Las funciones de las bases de datos gestionan principalmente las transformaciones de datos, los cálculos y las operaciones de búsqueda. Suelen estar integradas en varios procedimientos y aplicaciones. Por ejemplo, una organización sanitaria puede utilizar funciones para normalizar los datos de los pacientes o buscar códigos médicos.

Cada categoría representa diferentes aspectos de la lógica empresarial que está integrada en la capa de base de datos. Debe evaluar y planificar cuidadosamente cada uno de ellos para migrarlos a la capa de aplicación.

Durante esta fase de análisis, los clientes suelen enfrentarse a tres desafíos importantes. En primer lugar, las dependencias complejas surgen a través de llamadas a procedimientos anidadas, referencias entre esquemas y dependencias de datos implícitas. En segundo lugar, la gestión de las transacciones se vuelve fundamental, especialmente cuando se trata de transacciones de varios pasos y se mantiene la coherencia de los datos en los sistemas distribuidos. En tercer lugar, las consideraciones de rendimiento deben evaluarse cuidadosamente, especialmente en el caso de las operaciones de procesamiento por lotes, las actualizaciones masivas de datos y los cálculos en tiempo real, que actualmente se benefician de estar cerca de los datos.

Para abordar estos desafíos de manera efectiva, puede usar AWS Schema Conversion Tool (AWS SCT) para el análisis inicial y luego usar herramientas detalladas de mapeo de dependencias. Este enfoque le ayuda a comprender todo el alcance de la lógica de su base de datos y a crear una estrategia de migración integral que mantenga la continuidad empresarial durante la descomposición.

Al comprender a fondo estos componentes y desafíos, podrá planificar mejor su proceso de modernización y tomar decisiones informadas sobre qué elementos priorizar durante la migración a una arquitectura basada en microservicios.

Al analizar los componentes del código de la base de datos, cree una documentación completa para cada procedimiento, desencadenante y función almacenados. Comience por describir claramente su propósito y funcionalidad principal, incluidas las reglas comerciales que implementa. Detalle todos los parámetros de entrada y salida y anote sus tipos de datos y rangos válidos. Planifique las dependencias con respecto a otros objetos de la base de datos, sistemas externos y procesos posteriores. Defina claramente los límites de las transacciones y los requisitos de aislamiento para mantener la integridad de los datos. Documente cualquier expectativa de rendimiento, incluidos los requisitos de tiempo de respuesta y los patrones de utilización de los recursos. Por último, analice los patrones de uso para comprender los picos de carga, la frecuencia de ejecución y los períodos comerciales críticos.

Fase 2: Clasificación de la lógica empresarial

La descomposición efectiva de las bases de datos requiere una categorización sistemática de la lógica de la base de datos en sus dimensiones clave: complejidad, impacto en el negocio, dependencias, patrones de uso y dificultad de migración. Esta clasificación le ayuda a identificar los componentes de alto riesgo, determinar los requisitos de prueba y establecer las prioridades de migración. Por ejemplo, los procedimientos almacenados complejos con un alto impacto empresarial y un uso frecuente requieren una planificación cuidadosa y pruebas exhaustivas. Sin embargo, las funciones simples, poco utilizadas y con dependencias mínimas, pueden ser adecuadas para las primeras fases de migración.

Este enfoque estructurado crea una hoja de ruta de migración equilibrada que minimiza las interrupciones empresariales y, al mismo tiempo, mantiene la estabilidad del sistema. Al comprender estas interrelaciones, puede mejorar la secuencia de sus esfuerzos de descomposición y asignar los recursos de manera adecuada.

Fase 3: Migración de la lógica empresarial

Una vez que haya analizado y clasificado su lógica empresarial, es el momento de migrarla. Existen dos enfoques a la hora de migrar la lógica empresarial desde una base de datos monolítica: mover la lógica de la base de datos a la capa de aplicación o mover la lógica empresarial a otra base de datos que forme parte del microservicio.

Si migra la lógica empresarial a la aplicación, las tablas de la base de datos almacenan solo los datos y la base de datos no contiene ninguna lógica empresarial. Este es el enfoque recomendado. Puede usar Ispirer o herramientas de IA generativa, como Amazon Q Developer Kiro, o para convertir la lógica empresarial de la base de datos para la capa de aplicación, como la conversión a Java. Para obtener más información, consulte Migrar la lógica empresarial de la base de datos a la aplicación para acelerar la innovación y la flexibilidad (AWS entrada del blog).

Si migra la lógica empresarial a otra base de datos, puede utilizar AWS Schema Conversion Tool (AWS SCT) para convertir los esquemas de bases de datos y los objetos de código existentes en la base de datos de destino. Es compatible con servicios de AWS bases de datos diseñados específicamente, como Amazon DynamoDB, AmazonAurora y Amazon Redshift. Al proporcionar un informe de evaluación integral y capacidades de conversión automatizadas, AWS SCT ayuda a agilizar el proceso de transición y le permite centrarse en optimizar la nueva estructura de su base de datos para mejorar el rendimiento y la escalabilidad. A medida que avance en su proyecto de

modernización, AWS SCT podrá gestionar las conversiones incrementales para adoptar un enfoque gradual, lo que le permitirá validar y ajustar cada paso de la transformación de su base de datos.

Estrategia de reversión de la lógica empresarial

Dos aspectos fundamentales de cualquier estrategia de descomposición son mantener la compatibilidad con versiones anteriores e implementar procedimientos integrales de reversión. Estos elementos funcionan en conjunto para ayudar a proteger las operaciones durante el período de transición. En esta sección se describe cómo gestionar la compatibilidad durante el proceso de descomposición y establecer capacidades eficaces de reducción de emisiones en caso de emergencia que eviten posibles problemas.

Mantenga la compatibilidad con versiones anteriores

Durante la descomposición de la base de datos, es esencial mantener la compatibilidad con versiones anteriores para que las transiciones sean fluidas. Mantenga los procedimientos de bases de datos existentes de forma temporal e implemente gradualmente las nuevas funcionalidades. Utilice el control de versiones para realizar un seguimiento de todos los cambios y gestionar varias versiones de la base de datos simultáneamente. Planifique un período de coexistencia prolongado en el que tanto el sistema de origen como el de destino deban funcionar de forma fiable. Esto proporciona tiempo para probar y validar el nuevo sistema antes de retirar los componentes antiguos. Este enfoque minimiza las interrupciones de la actividad empresarial y proporciona una red de seguridad que se puede revertir en caso necesario.

Plan de reversión de emergencia

Una estrategia integral de reversión es esencial para una descomposición segura de la base de datos. Implemente indicadores de características en su código para controlar qué versión de la lógica empresarial está activa. Esto le permite cambiar instantáneamente entre las implementaciones nuevas y originales sin cambios en la implementación. Este enfoque proporciona un control detallado de la transición y le ayuda a revertirla rápidamente en caso de que surjan problemas. Mantenga la lógica original como una copia de seguridad verificada y mantenga procedimientos de reversión detallados que especifiquen los factores desencadenantes, las responsabilidades y los pasos de recuperación.

Pruebe periódicamente estos escenarios de reversión en diversas condiciones para validar su eficacia y asegúrese de que los equipos estén familiarizados con los procedimientos de emergencia. Los indicadores de características también permiten la implementación gradual al habilitar nuevas

funciones de forma selectiva para transacciones o grupos de usuarios específicos. Esto proporciona un nivel adicional de mitigación de riesgos durante la transición.

Desacoplar las relaciones de las tablas durante la descomposición de la base de datos

En esta sección se proporciona orientación sobre cómo desglosar las relaciones de tablas complejas y las operaciones de unión durante la descomposición monolítica de una base de datos. Una combinación de tablas combina filas de dos o más tablas en función de una columna relacionada entre ellas. El objetivo de separar estas relaciones es reducir el alto grado de acoplamiento entre tablas y, al mismo tiempo, mantener la integridad de los datos en todos los microservicios.

Esta sección contiene los siguientes temas:

Estrategia de desnormalización

La desnormalización es una estrategia de diseño de bases de datos que implica introducir intencionalmente redundancia mediante la combinación o duplicación de datos en las tablas. Al dividir una base de datos grande en bases de datos pequeñas, podría tener sentido duplicar algunos datos entre los servicios. Por ejemplo, almacenar los detalles básicos de los clientes, como el nombre y las direcciones de correo electrónico, tanto en un servicio de marketing como en un servicio de pedidos elimina la necesidad de realizar búsquedas constantes entre servicios. El servicio de marketing puede necesitar las preferencias de los clientes y la información de contacto para la segmentación de las campañas, mientras que el servicio de pedidos requiere los mismos datos para procesar los pedidos y las notificaciones. Si bien esto crea cierta redundancia de datos, puede mejorar considerablemente el rendimiento y la independencia del servicio, lo que permite al equipo de marketing gestionar sus campañas sin tener que depender de las consultas de atención al cliente en tiempo real.

Al implementar la desnormalización, céntrese en los campos de acceso frecuente que identifique mediante un análisis cuidadoso de los patrones de acceso a los datos. Puede utilizar herramientas,

como Oracle AWR informes opg_stat_statements, para comprender qué datos se recuperan habitualmente juntos. Los expertos en el campo también pueden proporcionar información valiosa sobre las agrupaciones de datos naturales. Recuerde que la desnormalización no es un all-or-nothing enfoque, sino solo datos duplicados que, de forma demostrable, mejoran el rendimiento del sistema o reducen las dependencias complejas.

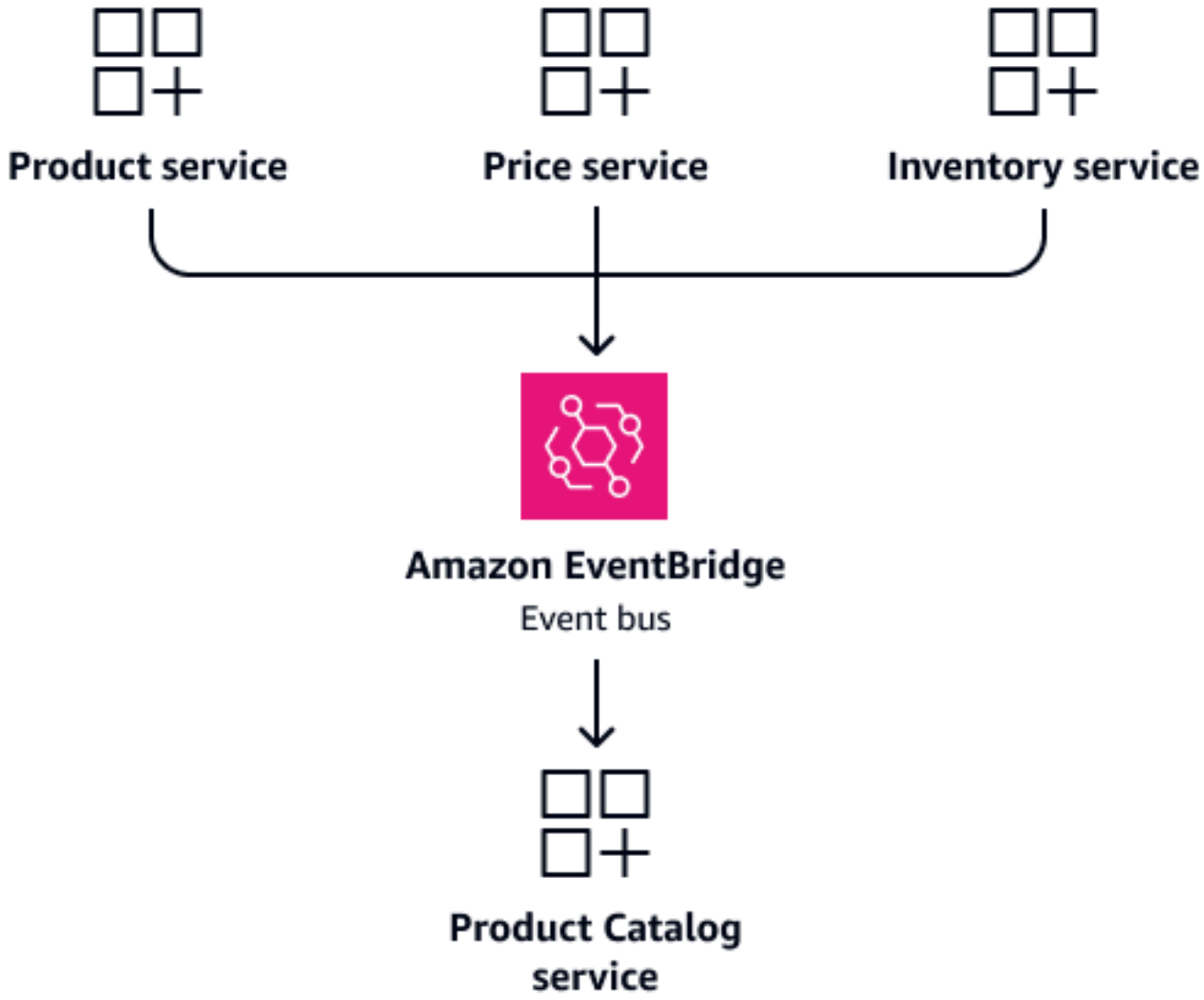
Reference-by-key estrategia

Una reference-by-key estrategia es un patrón de diseño de base de datos en el que las relaciones entre las entidades se mantienen mediante claves únicas en lugar de almacenar los datos relacionados reales. En lugar de las relaciones tradicionales de clave externa, los microservicios modernos suelen almacenar solo los identificadores únicos de los datos relacionados. Por ejemplo, en lugar de mantener todos los detalles del cliente en la tabla de pedidos, el servicio de pedidos solo almacena el identificador del cliente y recupera información adicional del cliente mediante una llamada a la API cuando es necesario. Este enfoque mantiene la independencia del servicio y, al mismo tiempo, garantiza el acceso a los datos relacionados.

Patrón CQRS

El patrón de segregación de responsabilidades de consultas de comandos (CQRS) separa las operaciones de lectura y escritura de un almacén de datos. Este patrón es particularmente útil en sistemas complejos con requisitos de alto rendimiento, especialmente aquellos con cargas asimétricas read/write . Si su aplicación necesita con frecuencia datos combinados de varias fuentes, puede crear un modelo CQRS dedicado en lugar de uniones complejas. Por ejemplo, en lugar de unir `Product` `Inventory` tablas y tablas en cada solicitud, mantenga una `Product Catalog` tabla consolidada que contenga los datos necesarios. `Pricing` Los beneficios de este enfoque pueden superar los costos de la tabla adicional.

Considere un escenario en el que `ProductPrice`, y `Inventory` los servicios necesiten con frecuencia información sobre el producto. En lugar de configurar estos servicios para acceder directamente a las tablas compartidas, cree un `Product Catalog` servicio dedicado. Este servicio mantiene su propia base de datos que contiene la información consolidada del producto. Actúa como una fuente única de información fiable para las consultas relacionadas con los productos. Cuando los detalles del producto, los precios o los niveles de inventario cambian, los servicios respectivos pueden publicar eventos para actualizar el `Product Catalog` servicio. Esto proporciona coherencia en los datos y, al mismo tiempo, mantiene la independencia del servicio. La siguiente imagen muestra esta configuración, en la que Amazon EventBridge actúa como bus de eventos.



Como se explica en Sincronización de datos basada en eventos la siguiente sección, mantenga el modelo CQRS actualizado durante los eventos. Cuando los detalles del producto, los precios o los niveles de inventario cambian, los servicios respectivos publican los eventos. El `Product Catalog` servicio se suscribe a estos eventos y actualiza su vista consolidada. Esto proporciona lecturas rápidas sin uniones complejas y mantiene la independencia del servicio.

Sincronización de datos basada en eventos

La sincronización de datos basada en eventos es un patrón en el que los cambios en los datos se capturan y propagan como eventos, lo que permite que diferentes sistemas o componentes mantengan los estados de los datos sincronizados. Cuando los datos cambien, en lugar de actualizar inmediatamente todas las bases de datos relacionadas, publique un evento para notificar a los

servicios suscritos. Por ejemplo, cuando un cliente cambia su dirección de envío en el `Customer` servicio, un `CustomerUpdated` evento inicia las actualizaciones del `Order` servicio y del `Delivery` servicio según la programación de cada servicio. Este enfoque reemplaza las uniones rígidas de tablas por actualizaciones flexibles y escalables basadas en eventos. Es posible que algunos servicios tengan datos desactualizados durante un período breve, pero la compensación es una mejor escalabilidad del sistema y una mayor independencia del servicio.

Implementar alternativas a las uniones de tablas

Comience la descomposición de la base de datos con operaciones de lectura, ya que suelen ser más fáciles de migrar y validar. Una vez que las rutas de lectura estén estables, aborde las operaciones de escritura más complejas. Para requisitos críticos de alto rendimiento, considere la posibilidad de implementar el patrón CQRS. Utilice una base de datos independiente y optimizada para las lecturas y mantenga otra para las escrituras.

Cree sistemas resilientes añadiendo lógica de reintento para las llamadas entre servicios e implementando las capas de almacenamiento en caché adecuadas. Supervise de cerca las interacciones de los servicios y configure alertas para detectar problemas de coherencia de los datos. El objetivo final no es una coherencia perfecta en todas partes, sino crear servicios independientes que funcionen bien y, al mismo tiempo, mantengan una precisión de datos aceptable para las necesidades de su empresa.

La naturaleza disociada de los microservicios introduce las siguientes nuevas complejidades en la administración de datos:

- Los datos se distribuyen. Los datos ahora residen en bases de datos independientes, que son administradas por servicios independientes.
- La sincronización en tiempo real entre los servicios no suele ser práctica y, en última instancia, requiere un modelo de coherencia.
- Las operaciones que antes se realizaban dentro de una sola transacción de base de datos ahora abarcan varios servicios.

Para hacer frente a estos desafíos, haga lo siguiente:

- Implemente una arquitectura basada en eventos: utilice las colas de mensajes y la publicación de eventos para propagar los cambios en los datos entre los servicios. Para obtener más información, consulte Creación de arquitecturas impulsadas por eventos en entornos sin servidores.

- Adopte el patrón de orquestación tipo saga: este patrón le ayuda a gestionar las transacciones distribuidas y a mantener la integridad de los datos en todos los servicios. Para obtener más información, consulte Crear una aplicación distribuida sin servidor utilizando un patrón de orquestación tipo saga en blogs. AWS
- Diseñe pensando en los errores: incorpore mecanismos de reintento, disyuntores y transacciones de compensación para gestionar los problemas de la red o los fallos del servicio.
- Utilice el sello de versiones: realice un seguimiento de las versiones de los datos para gestionar los conflictos y asegurarse de que se apliquen las actualizaciones más recientes.
- Conciliación periódica: implemente procesos de sincronización de datos periódicos para detectar y corregir cualquier incoherencia.

Ejemplo basado en escenarios

El siguiente ejemplo de esquema tiene dos tablas, una `Customer` tabla y una tabla: `Order`

```
-- Customer table
CREATE TABLE customer (
    customer_id INT PRIMARY KEY,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    email VARCHAR(255),
    phone VARCHAR(20),
    address TEXT,
    created_at TIMESTAMP
);

-- Order table
CREATE TABLE order (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date TIMESTAMP,
    total_amount DECIMAL(10,2),
    status VARCHAR(50),
    FOREIGN KEY (customer_id) REFERENCES customers(id)
);
```

El siguiente es un ejemplo de cómo se puede utilizar un enfoque desnormalizado:

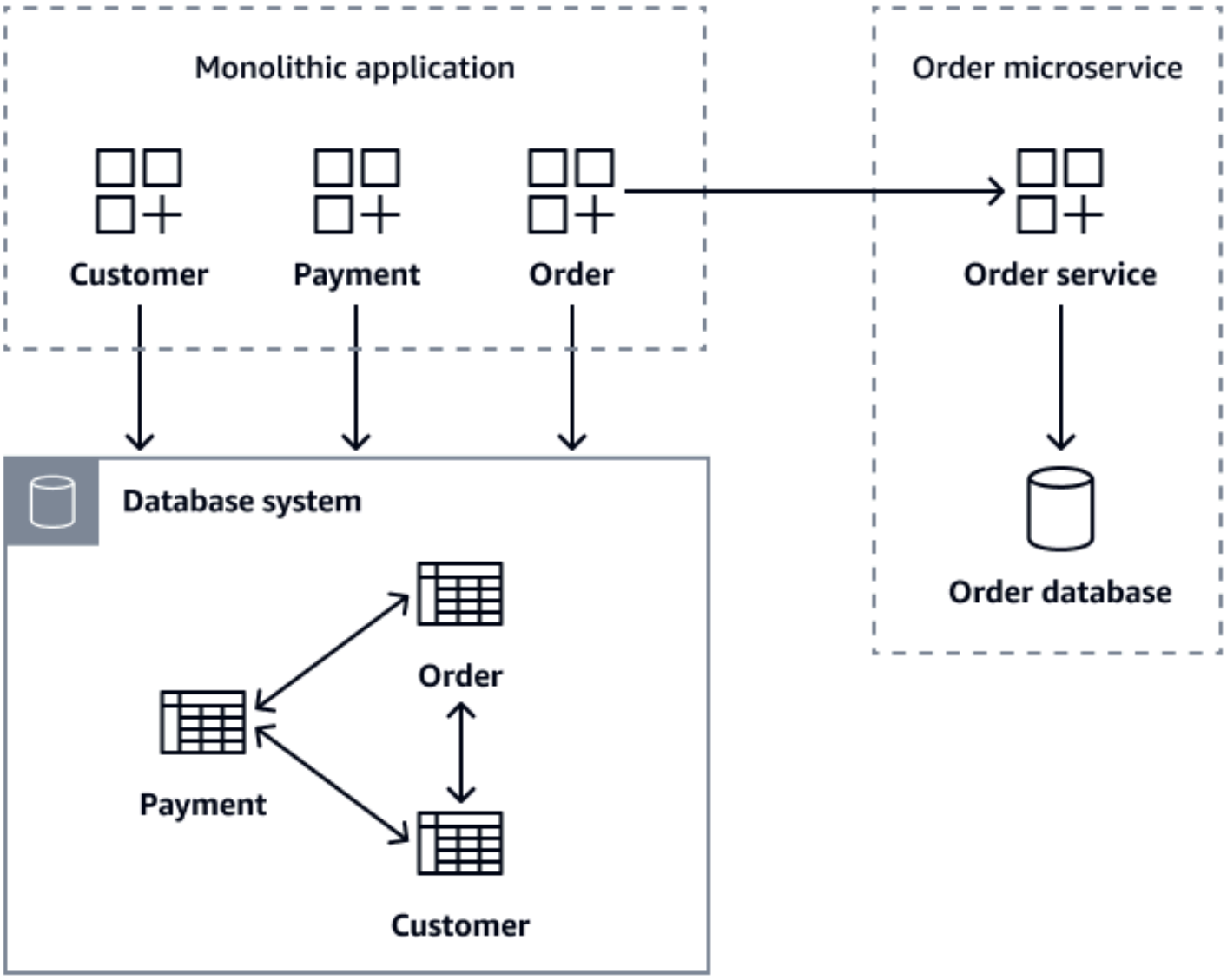
```
CREATE TABLE order (
```

```
    order_id INT PRIMARY KEY,
    customer_id INT,                  -- Reference only
    customer_first_name VARCHAR(100), -- Denormalized
    customer_last_name VARCHAR(100),  -- Denormalized
    customer_email VARCHAR(255),      -- Denormalized
    order_date TIMESTAMP,
    total_amount DECIMAL(10,2),
    status VARCHAR(50)
);
```

La nueva `Order` tabla tiene el nombre del cliente y las direcciones de correo electrónico que están desnormalizados. `customer_id`Se hace referencia a y no hay ninguna restricción de clave externa en la tabla. `Customer` Los beneficios de este enfoque desnormalizado son los siguientes:

- El `Order` servicio puede mostrar el historial de pedidos con los detalles del cliente y no requiere llamadas a la API al microservicio. `Customer`
- Si el `Customer` servicio no funciona, sigue siendo completamente funcional. `Order`
- Las consultas para el procesamiento de pedidos y la elaboración de informes se ejecutan más rápido.

El siguiente diagrama muestra una aplicación monolítica que recupera los datos de los pedidos mediante `getOrder(customer_id)` llamadas a la `createOrder(Order order)` API y a la API al `Order` microservicio. `getOrder(order_id)` `getCustomerOders(customer_id)`



Durante la migración de los microservicios, puede mantener la `Order` tabla en la base de datos monolítica como medida de seguridad transitoria y garantizar que la aplicación antigua siga funcionando. Sin embargo, es fundamental que todas las nuevas operaciones relacionadas con los pedidos se envíen a través de la API de `Order` microservicios, que mantiene su propia base de datos y, al mismo tiempo, graba en la base de datos antigua como respaldo. Este patrón de escritura doble proporciona una red de seguridad. Permite una migración gradual y, al mismo tiempo, mantiene la estabilidad del sistema. Una vez que todos los clientes hayan migrado correctamente al nuevo microservicio, puede dejar obsoleta la `Order` tabla antigua de la base de datos monolítica. Tras descomponer la aplicación monolítica y su base de datos en microservicios independientes `Customer` y `Order` microservicios, mantener la coherencia de los datos se convierte en el principal desafío.

Mejores prácticas para la descomposición de bases de datos

Al descomponer una base de datos monolítica, las organizaciones deben establecer marcos claros para hacer un seguimiento del progreso, mantener el conocimiento del sistema y abordar los desafíos emergentes. Esta sección proporciona las mejores prácticas para medir el éxito de la descomposición, mantener la documentación crucial, implementar procesos de mejora continua y afrontar los desafíos más comunes. Comprender y seguir estas pautas le ayudará a asegurarse de que las iniciativas de descomposición de las bases de datos generen los beneficios esperados y, al mismo tiempo, minimicen las interrupciones operativas y la deuda técnica.

Esta sección contiene los siguientes temas:

- Medir el éxito
- Requisitos de documentación
- Estrategia de mejora continua
- Superar los desafíos comunes en la descomposición de las bases de datos

Medir el éxito

Realice un seguimiento del éxito de la descomposición mediante una combinación de métricas técnicas, operativas y empresariales. Técnicamente, supervise los tiempos de respuesta a las consultas, las mejoras en el tiempo de actividad del sistema y el aumento de la frecuencia de implementación. Desde el punto de vista operativo, mida la reducción de incidentes, la velocidad de resolución de problemas y las mejoras en la utilización de los recursos. Para el desarrollo, controle la velocidad de implementación de las funciones, la aceleración del ciclo de lanzamiento y la reducción de las dependencias entre equipos. El impacto empresarial debería traducirse en una reducción de los costes operativos, una mayor rapidez time-to-market y una mayor satisfacción de los clientes. Estas métricas suelen definirse durante la fase de alcance. Para obtener más información, consulte Definir el alcance y los requisitos de la descomposición de las bases de datos en esta guía.

Requisitos de documentación

Mantenga la documentación de la arquitectura up-to-date del sistema con límites de servicio, flujos de datos y especificaciones de interfaz claros. Utilice los registros de decisiones de arquitectura

(ADRs) para recopilar las decisiones técnicas clave, incluidos su contexto, las consecuencias y las alternativas consideradas. Por ejemplo, documente por qué se separaron primero servicios específicos o cómo se hicieron ciertas concesiones en cuanto a la coherencia de los datos.

Programe revisiones mensuales de la arquitectura para evaluar el estado del sistema a través de métricas clave: las tendencias de rendimiento, el cumplimiento de las normas de seguridad y las dependencias entre servicios. Incluya los comentarios de los equipos de desarrollo sobre los desafíos de integración y los problemas operativos. Este ciclo de revisión regular le ayuda a identificar los problemas emergentes con antelación y confirma que los esfuerzos de descomposición siguen alineados con los objetivos empresariales.

Estrategia de mejora continua

Trate la descomposición de la base de datos como un proceso iterativo, no como un proyecto único. Supervise las métricas de rendimiento del sistema y las interacciones de los servicios para identificar las oportunidades de optimización. Cada trimestre, priorice la solución de la deuda técnica en función del impacto operativo y los costos de mantenimiento. Por ejemplo, automatice las operaciones de bases de datos que se realizan con frecuencia, mejore la cobertura de monitoreo y refine los procedimientos de implementación en función de los patrones aprendidos.

Superar los desafíos comunes en la descomposición de las bases de datos

La optimización del rendimiento requiere un enfoque multifacético. Implemente el almacenamiento en caché estratégico en los límites del servicio, optimice los patrones de consulta en función del uso real y supervise continuamente las métricas clave. Aborde los cuellos de botella en el rendimiento de forma proactiva analizando las tendencias y estableciendo umbrales claros para la intervención.

Los desafíos de consistencia de los datos exigen decisiones arquitectónicas cuidadosas. Implemente patrones basados en eventos para las actualizaciones entre servicios y utilice patrones de orquestación tipo saga para transacciones complejas. Defina límites de servicio claros y acepte la coherencia eventual cuando los requisitos empresariales lo permitan. Este equilibrio entre la coherencia y la autonomía del servicio es crucial para el éxito de la descomposición.

La excelencia operativa requiere la automatización de las tareas rutinarias y los procedimientos estandarizados en todos los servicios. Mantenga una supervisión exhaustiva con umbrales de alerta claros e invierta en la formación periódica del equipo para aprender nuevos patrones y herramientas.

Este enfoque sistemático de las operaciones promueve una prestación de servicios fiable y, al mismo tiempo, gestiona la complejidad.

Preguntas frecuentes sobre la descomposición de bases de datos

Esta completa sección de preguntas frecuentes aborda las preguntas y los desafíos más comunes a los que se enfrentan las organizaciones al emprender proyectos de descomposición de bases de datos. Desde la definición del alcance y los requisitos iniciales hasta la migración de los procedimientos almacenados, estas preguntas proporcionan información práctica y enfoques estratégicos para ayudar a los equipos a emprender con éxito su proceso de modernización de las bases de datos. Tanto si se encuentra en la fase de planificación como si ya está ejecutando su estrategia de descomposición, estas respuestas pueden ayudarle a evitar los errores más comunes e implementar las mejores prácticas para obtener resultados óptimos.

Esta sección contiene los siguientes temas:

FAQs sobre la definición del alcance y los requisitos

En esta Definir el alcance y los requisitos para la descomposición de la base de datos sección de esta guía se explica cómo analizar las interacciones, mapear las dependencias y establecer los criterios de éxito. Esta sección de preguntas frecuentes aborda cuestiones clave sobre el establecimiento y la gestión de los límites de los proyectos. Ya sea que se trate de limitaciones técnicas poco claras, de necesidades departamentales contradictorias o de requisitos empresariales en evolución, estas FAQs proporcionan una guía práctica para mantener un enfoque equilibrado.

Esta sección contiene las siguientes preguntas:

- ¿Cómo puedo equilibrar las necesidades empresariales inmediatas con los objetivos técnicos a largo plazo?
- ¿Cómo me aseguro de no incumplir los requisitos críticos de las partes interesadas silenciosas?
- ¿Se aplican estas recomendaciones a las bases de datos monolíticas de mainframe?

¿Qué tan detallada debe ser la definición inicial del alcance?

Partiendo de las necesidades de sus clientes, defina el alcance del proyecto con suficiente detalle como para identificar los límites del sistema y las dependencias críticas y, al mismo tiempo, mantenga la flexibilidad de descubrimiento. Mapee los elementos esenciales, incluidas las interfaces del sistema, las partes interesadas clave y las principales limitaciones técnicas. Comience poco a poco seleccionando una parte del sistema limitada y de bajo riesgo que proporcione un valor mensurable. Este enfoque ayuda a los equipos a aprender y ajustar las estrategias antes de abordar componentes más complejos.

Documente los requisitos empresariales fundamentales que impulsan el esfuerzo de descomposición, pero evite especificar en exceso los detalles que podrían cambiar durante la implementación. Este enfoque equilibrado garantiza que los equipos puedan avanzar con claridad y, al mismo tiempo, adaptarse a los nuevos conocimientos y desafíos que surjan durante el proceso de modernización.

¿Qué sucede si descubro dependencias adicionales después de iniciar el proyecto?

Espere descubrir dependencias adicionales a medida que avance el proyecto. Mantenga un registro de dependencias activo y lleve a cabo revisiones periódicas del alcance para evaluar el impacto en los plazos y los recursos. Implemente un proceso de gestión de cambios claro e incluya tiempo de reserva en los planes de los proyectos para gestionar los descubrimientos inesperados. El objetivo no es evitar los cambios, sino gestionarlos de forma eficaz. Esto ayuda a los equipos a adaptarse rápidamente y, al mismo tiempo, a mantener el impulso del proyecto.

¿Cómo trato a las partes interesadas de los diferentes departamentos que tienen requisitos contradictorios?

Gestione los requisitos departamentales contradictorios mediante una priorización clara que se base en el valor empresarial y el impacto en el sistema. Consiga el patrocinio de los ejecutivos para tomar

decisiones clave y resolver los conflictos rápidamente. Programe reuniones periódicas de alineación de las partes interesadas para analizar las ventajas y desventajas y mantener la transparencia. Documente todas las decisiones y sus motivos para promover una comunicación clara y mantener el impulso del proyecto. Centra los debates en los beneficios empresariales cuantificables y no en las preferencias departamentales.

¿Cuál es la mejor manera de evaluar las limitaciones técnicas cuando la documentación es deficiente o está desactualizada?

Cuando la documentación sea deficiente, combine el análisis tradicional con las herramientas de IA modernas. Utilice modelos de lenguaje de gran tamaño (LLMs) para analizar los repositorios de código, los registros y la documentación existente a fin de identificar los patrones y las posibles limitaciones. Entreviste a desarrolladores y arquitectos de bases de datos con experiencia para validar los hallazgos de la IA y descubrir las limitaciones no documentadas. Implemente herramientas de monitoreo que tengan capacidades de inteligencia artificial mejoradas para observar el comportamiento del sistema y predecir posibles problemas.

Cree pequeños experimentos técnicos que validen sus suposiciones. Puede utilizar herramientas de prueba impulsadas por IA para acelerar el proceso. Documente los hallazgos en una base de conocimientos que pueda mejorarse continuamente mediante actualizaciones asistidas por IA. Considere la posibilidad de contratar a expertos en la materia para áreas complejas y utilice herramientas de programación por pares de IA para acelerar sus esfuerzos de análisis y documentación.

¿Cómo puedo equilibrar las necesidades empresariales inmediatas con los objetivos técnicos a largo plazo?

Cree una hoja de ruta del proyecto por fases que alinee las necesidades empresariales inmediatas con los objetivos técnicos a largo plazo. Identifique las ventajas rápidas que generen un valor tangible desde el principio, de modo que pueda fomentar la confianza de las partes interesadas. Divida la descomposición en hitos claros. Cada uno de ellos debería ofrecer beneficios empresariales cuantificables y, al mismo tiempo, avanzar hacia los objetivos arquitectónicos. Mantenga la flexibilidad para abordar las necesidades empresariales urgentes mediante revisiones y ajustes periódicos de la hoja de ruta.

¿Cómo me aseguro de no incumplir los requisitos críticos de las partes interesadas silenciosas?

Haga un mapa de todas las posibles partes interesadas de la organización, incluidos los propietarios de los sistemas intermedios y los usuarios indirectos. Cree múltiples canales de retroalimentación mediante entrevistas estructuradas, talleres y sesiones de revisión periódicas. Cree proof-of-concepts prototipos para hacer que los requisitos sean tangibles y suscite debates significativos. Por ejemplo, un panel sencillo que muestre las dependencias del sistema suele revelar partes interesadas y requisitos ocultos que inicialmente no eran evidentes.

Lleve a cabo sesiones de validación periódicas con las partes interesadas que se expresen y no hablen, y asegúrese de que se capten todas las perspectivas. Las opiniones críticas suelen provenir de las personas más cercanas a las operaciones diarias, y no de las voces más ruidosas en las reuniones de planificación.

¿Se aplican estas recomendaciones a las bases de datos monolíticas de mainframe?

La metodología descrita en esta guía también se aplica a la descomposición de bases de datos monolíticas de mainframe. Los principales desafíos de estas bases de datos son gestionar los requisitos de las distintas partes interesadas. Las recomendaciones tecnológicas de esta guía podrían aplicarse a las bases de datos monolíticas de mainframe. Si el ordenador central tiene una base de datos relacional, como una base de datos de procesamiento de transacciones en línea (OLTP), se aplican muchas de las recomendaciones. En el caso de las bases de datos de procesamiento analítico en línea (OLAP), como las que se utilizan para generar informes empresariales, solo se aplican algunas de las recomendaciones.

FAQs sobre el control del acceso a las bases de

En la Control del acceso a la base de datos durante la descomposición sección de esta guía se explica cómo controlar el acceso a las bases de datos mediante el patrón de servicio de empaquetado de bases de datos. En esta sección de preguntas frecuentes se abordan las inquietudes y preguntas más frecuentes sobre la introducción de un servicio de empaquetado de bases de datos, como su posible impacto en el rendimiento, la gestión de los procedimientos almacenados existentes, la gestión de transacciones complejas y la supervisión de los cambios de esquema.

En esta sección se incluyen las siguientes preguntas:

¿No se convertirá el servicio de embalaje en un nuevo cuello de botella?

Si bien el servicio de empaquetado de bases de datos añade un salto de red adicional, el impacto suele ser mínimo. Puede escalar el servicio de forma horizontal y, por lo general, las ventajas del acceso controlado superan el reducido costo de rendimiento. Considérelo una compensación temporal entre el rendimiento y la capacidad de mantenimiento.

¿Qué ocurre con los procedimientos almacenados existentes?

Inicialmente, el servicio contenedor de bases de datos puede exponer los procedimientos almacenados como métodos de servicio. Con el tiempo, puede trasladar gradualmente la lógica a la capa de aplicación, lo que mejora las pruebas y el control de versiones. Migre la lógica empresarial de forma gradual para minimizar el riesgo.

¿Cómo gestiono los cambios de esquema durante la transición?

Centralice el control de los cambios de esquema a través del equipo de servicio de empaquetado. Este equipo es responsable de mantener una visibilidad integral entre todos los consumidores. Este equipo revisa los cambios propuestos para determinar su impacto en todo el sistema, coordina con los equipos afectados e implementa las modificaciones mediante un proceso de despliegue controlado. Por ejemplo, al añadir nuevos campos, este equipo debe mantener la compatibilidad con versiones anteriores, implementando valores predeterminados o permitiendo inicialmente los valores nulos.

Establezca un proceso de gestión de cambios claro que incluya la evaluación del impacto, los requisitos de prueba y los procedimientos de reversión. Utilice herramientas de control de versiones de bases de datos y mantenga una documentación clara de todos los cambios. Este enfoque centralizado evita que las modificaciones del esquema interrumpan los servicios dependientes y mantiene la estabilidad del sistema.

FAQs sobre el análisis de la cohesión y el acoplamiento

Comprender y analizar eficazmente el acoplamiento y la cohesión de las bases de datos es fundamental para una descomposición exitosa de las bases de datos. El acoplamiento y la cohesión

se analizan en la Análisis de la cohesión y el acoplamiento para la descomposición de la base de datos sección de esta guía. Esta sección de preguntas frecuentes aborda cuestiones clave sobre la identificación de los niveles de granularidad adecuados, la selección de las herramientas de análisis adecuadas, la documentación de los hallazgos y la priorización de los problemas de acoplamiento.

En esta sección se incluyen las siguientes preguntas:

- ¿Cómo identifico el nivel correcto de granularidad al analizar el acoplamiento?
- ¿Qué herramientas puedo usar para analizar el acoplamiento y la cohesión de las bases de datos?
- ¿Cuál es la mejor manera de documentar las conclusiones sobre el acoplamiento y la cohesión?
- ¿Cómo puedo priorizar qué cuestiones de acoplamiento abordar primero?
- ¿Cómo gestiono las transacciones que abarcan varias operaciones?

¿Cómo identifico el nivel correcto de granularidad al analizar el acoplamiento?

Comience con un análisis amplio de las relaciones de las bases de datos y, a continuación, profundice sistemáticamente para identificar los puntos de separación naturales. Utilice las herramientas de análisis de bases de datos para mapear las relaciones a nivel de tabla, las dependencias de los esquemas y los límites de las transacciones. Por ejemplo, examine los patrones de unión en las consultas de SQL para comprender las dependencias de acceso a los datos. También puede analizar los registros de transacciones para identificar los límites de los procesos empresariales.

Concéntrese en las áreas en las que el acoplamiento es mínimo por naturaleza. A menudo, se alinean con los límites del dominio empresarial y representan puntos de descomposición óptimos. Al determinar los límites de servicio adecuados, tenga en cuenta tanto el acoplamiento técnico (como las tablas compartidas y las claves externas) como el acoplamiento empresarial (como los flujos de procesos y las necesidades de presentación de informes).

¿Qué herramientas puedo usar para analizar el acoplamiento y la cohesión de las bases de datos?

Puede utilizar una combinación de herramientas automatizadas y análisis manuales para evaluar el acoplamiento y la cohesión de las bases de datos. Las siguientes herramientas pueden ayudarle con esta evaluación:

- Herramientas de visualización de esquemas: puede usar herramientas como SchemaSpyo pgAdminpara generar diagramas ER. Estos diagramas revelan las relaciones de las tablas y los posibles puntos de acoplamiento.
- Herramientas de análisis de consultas: puede utilizar pg_stat_statementso SQL Server Query Storeidentificar tablas y patrones de acceso que se unen con frecuencia.
- Herramientas de creación de perfiles de bases de datos: herramientas como Oracle SQL Developero que MySQL Workbenchproporcionan información sobre el rendimiento de las consultas y las dependencias de los datos.
- Herramientas de mapeo de dependencias: las AWS Schema Conversion Tool (AWS SCT) pueden ayudarle a visualizar las relaciones entre esquemas e identificar componentes estrechamente acoplados. vFunctionpuede ayudarlo a identificar los límites del dominio mediante el análisis de los límites funcionales y de dominio de la aplicación.
- Herramientas de supervisión de transacciones: puede utilizar herramientas específicas de bases de datos, como Oracle Enterprise Manager o SQL Server Extended Events, para analizar los límites de las transacciones.
- Herramientas de migración de lógica empresarial: puede utilizar Ispirerherramientas de IA generativas, como Amazon Q Developer Kiro, o para convertir la lógica empresarial de la base de datos para la capa de aplicación, como la conversión a Java.

Combine estos análisis automatizados con la revisión manual de los procesos empresariales y el conocimiento del dominio para comprender completamente la combinación de sistemas. Este enfoque multifacético garantiza que en su estrategia de descomposición se tengan en cuenta tanto las perspectivas técnicas como las empresariales.

¿Cuál es la mejor manera de documentar las conclusiones sobre el acoplamiento y la cohesión?

Cree una documentación completa que visualice las relaciones de las bases de datos y los patrones de uso. Los siguientes son los tipos de activos que puede utilizar para registrar sus hallazgos:

- Matrices de dependencia: mapee las dependencias de las tablas y resalte las áreas de alto acoplamiento.
- Diagramas de relaciones: utilice los diagramas ER para mostrar las conexiones de los esquemas y las relaciones de claves externas.

- Mapas térmicos de uso de tablas: visualice la frecuencia de las consultas y los patrones de acceso a los datos en todas las tablas.
- Diagramas de flujo de transacciones: documenta las transacciones de varias tablas y sus límites.
- Mapas de límites de dominio: describa los posibles límites de los servicios en función de los dominios comerciales.

Combine estos artefactos en un documento y actualícelo periódicamente a medida que avance la descomposición. Para los diagramas, puede utilizar herramientas como draw.ioo Lucidchart. Considere la posibilidad de implementar una wiki para facilitar el acceso y la colaboración en equipo. Este enfoque de documentación multifacético proporciona una comprensión clara y compartida del acoplamiento y la cohesión de los sistemas.

¿Cómo puedo priorizar qué cuestiones de acoplamiento abordar primero?

Priorice los problemas de acoplamiento basándose en una evaluación equilibrada de los factores comerciales y técnicos. Evalúe cada problema en función del impacto empresarial (como los ingresos y la experiencia del cliente), el riesgo técnico (como la estabilidad del sistema y la integridad de los datos), el esfuerzo de implementación y las capacidades del equipo. Cree una matriz de priorización que puntúe cada problema del 1 al 5 en estas dimensiones. Esta matriz le ayuda a identificar las oportunidades más valiosas con riesgos manejables.

Comience con cambios de alto impacto y bajo riesgo que se ajusten a la experiencia actual del equipo. Esto te ayuda a generar confianza en la organización y a impulsar cambios más complejos. Este enfoque promueve una ejecución realista y maximiza el valor empresarial. Revise y ajuste las prioridades periódicamente para ayudar a mantenerlas alineadas con las cambiantes necesidades empresariales y la capacidad del equipo.

¿Cómo gestiono las transacciones que abarcan varias operaciones?

Gestione las transacciones de múltiples operaciones mediante una coordinación de nivel de servicio cuidadosamente diseñada. Implemente patrones tipo saga para transacciones distribuidas complejas. Divídalos en pasos más pequeños y reversibles que se puedan gestionar de forma independiente. Por ejemplo, un flujo de procesamiento de pedidos puede dividirse en pasos separados para la verificación del inventario, el procesamiento de pagos y la creación de pedidos, cada uno con su propio mecanismo de compensación.

Siempre que sea posible, rediseñe las operaciones para que sean más atómicas, lo que reduce la necesidad de realizar transacciones distribuidas. Cuando las transacciones distribuidas sean

inevitables, implemente mecanismos sólidos de seguimiento y compensación para promover la coherencia de los datos. Supervise las tasas de finalización de transacciones e implemente procedimientos claros de recuperación de errores para mantener la confiabilidad del sistema.

FAQs sobre la migración de la lógica empresarial a la capa de aplicación

La migración de la lógica empresarial de la base de datos a la capa de aplicación es un aspecto crítico y complejo de la modernización de las bases de datos. Esta migración de la lógica empresarial se analiza en la Migración de la lógica empresarial de la base de datos a la capa de aplicación sección de esta guía. En esta sección de preguntas frecuentes se abordan las preguntas más frecuentes sobre la gestión eficaz de esta transición, desde la selección de los candidatos iniciales para la migración hasta la gestión de procedimientos almacenados y desencadenantes complejos.

En esta sección se incluyen las siguientes preguntas:

- ¿Cómo identifico qué procedimientos almacenados migrar primero?
- ¿Cuáles son los riesgos de trasladar la lógica a la capa de aplicación?
- ¿Cómo mantengo el rendimiento al alejar la lógica de la base de datos?
- ¿Qué debo hacer con los procedimientos almacenados complejos que incluyen varias tablas?
- ¿Cómo puedo gestionar los activadores de las bases de datos durante la migración?
- ¿Cuál es la mejor manera de probar la lógica empresarial migrada?
- ¿Cómo administro el período de transición cuando existen tanto la lógica de la base de datos como la de la aplicación?
- ¿Cómo puedo gestionar los escenarios de error en la capa de aplicación que antes gestionaba la base de datos?

¿Cómo identifico qué procedimientos almacenados migrar primero?

Comience por identificar los procedimientos almacenados que ofrezcan la mejor combinación de bajo riesgo y alto valor de aprendizaje. Concéntrese en los procedimientos que tengan dependencias mínimas, una funcionalidad clara y un impacto empresarial no crítico. Estos son los candidatos ideales para la migración inicial porque ayudan al equipo a generar confianza y a establecer patrones. Por ejemplo, elija procedimientos que gestionen operaciones de datos sencillas en lugar de aquellos que gestionen transacciones complejas o lógicas empresariales críticas.

Utilice las herramientas de supervisión de bases de datos para analizar los patrones de uso e identificar los procedimientos a los que se accede con poca frecuencia como primeros candidatos. Este enfoque minimiza el riesgo empresarial y, al mismo tiempo, proporciona una valiosa experiencia para abordar migraciones más complejas en el futuro. Puntúe cada procedimiento en función de la complejidad, la criticidad empresarial y los niveles de dependencia para crear una secuencia de migración priorizada.

¿Cuáles son los riesgos de trasladar la lógica a la capa de aplicación?

Mover la lógica de la base de datos a la capa de aplicación presenta varios desafíos clave. El rendimiento del sistema puede degradarse debido al aumento de las llamadas a la red, especialmente en el caso de operaciones con uso intensivo de datos que anteriormente se gestionaban en la base de datos. La administración de transacciones se hace más compleja y requiere una coordinación cuidadosa para mantener la integridad de los datos en todas las operaciones distribuidas. Garantizar la coherencia de los datos se convierte en un desafío, especialmente para las operaciones que antes dependían de restricciones a nivel de base de datos.

La posible interrupción del negocio durante la migración y la curva de aprendizaje para los desarrolladores también son preocupaciones importantes. Mitigue estos riesgos mediante una planificación minuciosa, pruebas exhaustivas en entornos por etapas y una migración gradual que comience con los componentes menos críticos. Implemente procedimientos sólidos de supervisión y reversión para identificar y abordar rápidamente los problemas de producción.

¿Cómo mantengo el rendimiento al alejar la lógica de la base de datos?

Implemente los mecanismos de almacenamiento en caché adecuados para los datos a los que se accede con frecuencia, optimice los patrones de acceso a los datos para minimizar las llamadas a la red y utilice el procesamiento por lotes para las operaciones masivas. En el caso de non-time-critical las operaciones, considere el procesamiento asíncrono para mejorar la capacidad de respuesta del sistema.

Supervise de cerca las métricas de rendimiento de las aplicaciones y ajústelas según sea necesario. Por ejemplo, puede sustituir varias operaciones de una sola fila por un procesamiento masivo, puede almacenar en caché los datos de referencia que cambian con poca frecuencia y puede optimizar los patrones de consulta para reducir la transferencia de datos. Las pruebas y los ajustes de rendimiento periódicos ayudan al sistema a mantener tiempos de respuesta aceptables y mejoran la capacidad de mantenimiento y la escalabilidad.

¿Qué debo hacer con los procedimientos almacenados complejos que incluyen varias tablas?

Acérquese a procedimientos almacenados complejos y con varias tablas mediante la descomposición sistemática. Comience por dividirlos en componentes más pequeños y coherentes desde el punto de vista lógico, e identifique claramente los límites de las transacciones y las dependencias de los datos. Cree interfaces de servicio para cada componente lógico. Esto le ayuda a migrar gradualmente sin interrumpir la funcionalidad existente.

Implemente una step-by-step migración, empezando por los componentes menos acoplados. En el caso de procedimientos muy complejos, considere la posibilidad de mantenerlos temporalmente en la base de datos y, al mismo tiempo, migrar las partes más simples. Este enfoque híbrido mantiene la estabilidad del sistema mientras usted avanza hacia sus objetivos arquitectónicos. Supervise continuamente el rendimiento y la funcionalidad durante la migración y prepárese para ajustar su estrategia en función de los resultados.

¿Cómo puedo gestionar los activadores de las bases de datos durante la migración?

Transforma los activadores de bases de datos en controladores de eventos a nivel de aplicación y, al mismo tiempo, mantiene la funcionalidad del sistema. Sustituya los activadores sincrónicos por patrones basados en eventos que envíen mensajes a las colas para realizar operaciones asincrónicas. Considere la posibilidad de utilizar Amazon Simple Notification Service (Amazon SNS) o Amazon Simple Queue Service (Amazon SQS) para las colas de mensajes. Para cumplir con los requisitos de auditoría, implemente el registro a nivel de aplicación o utilice las funciones de captura de datos de cambios en la base de datos (CDC).

Analice el propósito y la importancia de cada desencadenante. Algunos desencadenantes podrían funcionar mejor con la lógica de la aplicación, y otros podrían requerir patrones de origen de eventos para mantener la coherencia de los datos. Comience con activadores simples, como los registros de auditoría, antes de abordar los más complejos que gestionan las reglas empresariales o la integridad de los datos. Supervise cuidadosamente durante la migración para asegurarse de que no se pierda la funcionalidad o la coherencia de los datos.

¿Cuál es la mejor manera de probar la lógica empresarial migrada?

Implemente un enfoque de pruebas de varios niveles antes de implementar la lógica empresarial migrada. Comience con pruebas unitarias para el nuevo código de aplicación y, a continuación,

añada pruebas de integración que abarquen los flujos end-to-end empresariales. Ejecute las implementaciones antiguas y nuevas en paralelo y, a continuación, compare los resultados para validar la equivalencia funcional. Realice pruebas de rendimiento en diversas condiciones de carga para comprobar que el comportamiento del sistema coincide o supera las capacidades anteriores.

Utilice indicadores de características para controlar el despliegue y poder revertirlo rápidamente en caso de que surjan problemas. Involucre a los usuarios empresariales en la validación, especialmente en el caso de los flujos de trabajo críticos. Supervise las métricas clave durante la implementación inicial y aumente gradualmente el tráfico hacia la nueva implementación. En todo momento, mantenga la capacidad de volver a la lógica original de la base de datos si es necesario.

¿Cómo administro el período de transición cuando existen tanto la lógica de la base de datos como la de la aplicación?

Cuando la lógica de la base de datos y la de la aplicación estén en uso, implemente indicadores de funciones que controlen el flujo de tráfico y permitan cambiar rápidamente entre las implementaciones antiguas y las nuevas. Mantenga un control riguroso de las versiones y documente con claridad tanto las implementaciones como sus responsabilidades respectivas. Configure una supervisión integral de ambos sistemas para identificar rápidamente cualquier discrepancia o problema de rendimiento.

Establezca procedimientos de reversión claros para cada componente migrado, de modo que pueda volver a la lógica original si es necesario. Comuníquese periódicamente con todas las partes interesadas sobre el estado de la transición, los posibles impactos y los procedimientos de escalamiento. Este enfoque le ayuda a migrar gradualmente y, al mismo tiempo, a mantener la estabilidad del sistema y la confianza de las partes interesadas.

¿Cómo puedo gestionar los escenarios de error en la capa de aplicación que antes gestionaba la base de datos?

Sustituya la gestión de errores a nivel de base de datos por mecanismos sólidos a nivel de aplicación. Implemente disyuntores y reintente la lógica para los fallos transitorios. Utilice transacciones de compensación para mantener la coherencia de los datos en todas las operaciones distribuidas. Por ejemplo, si se produce un error en la actualización de un pago, la aplicación debería volver a intentarlo automáticamente dentro de los límites definidos e iniciar acciones de compensación si fuera necesario.

Configure una supervisión y alertas exhaustivas para identificar rápidamente los problemas y mantenga registros de auditoría detallados para la resolución de problemas. Diseñe la gestión de errores de forma que sea lo más automatizada posible y defina vías de escalamiento claras para los escenarios que requieran la intervención humana. Este enfoque de varios niveles proporciona resiliencia al sistema y, al mismo tiempo, mantiene la integridad de los datos y la continuidad de los procesos empresariales.

Próximos pasos para la descomposición de la base de datos en AWS

Tras implementar las estrategias iniciales de descomposición de las bases de datos mediante servicios de empaquetado de bases de datos y trasladar la lógica empresarial a la capa de aplicaciones, las organizaciones deben planificar su próxima evolución. En esta sección se describen las consideraciones clave para continuar su proceso de modernización.

Esta sección contiene los siguientes temas:

- Estrategias incrementales para la descomposición de bases de datos
- Consideraciones técnicas para entornos de bases de datos distribuidas
- Cambios organizativos para dar soporte a las arquitecturas distribuidas

Estrategias incrementales para la descomposición de bases de datos

La descomposición de las bases de datos sigue una evolución gradual a través de tres fases distintas. Los equipos primero empaquetan la base de datos monolítica con un servicio de empaquetado de bases de datos para controlar el acceso. Luego, comienzan a dividir los datos en bases de datos específicas del servicio y, al mismo tiempo, mantienen la base de datos principal para satisfacer las necesidades heredadas. Por último, completan la migración de la lógica empresarial para realizar la transición a bases de datos de servicios totalmente independientes.

A lo largo de este proceso, los equipos deben implementar patrones de sincronización de datos cuidadosos y validar continuamente la coherencia de los servicios. La supervisión del rendimiento se vuelve crucial para identificar y abordar los posibles problemas de forma temprana. Dado que los servicios evolucionan de forma independiente, sus esquemas deben optimizarse en función de los patrones de uso reales, y es necesario eliminar las estructuras redundantes que se han ido acumulando a lo largo del tiempo.

Este enfoque gradual ayuda a minimizar los riesgos y, al mismo tiempo, a mantener la estabilidad del sistema durante todo el proceso de transformación.

Consideraciones técnicas para entornos de bases de datos distribuidas

En un entorno de bases de datos distribuidas, la supervisión del rendimiento se vuelve esencial para identificar y abordar los cuellos de botella de forma temprana. Los equipos deben implementar sistemas de monitoreo integrales y estrategias de almacenamiento en caché para mantener los niveles de rendimiento. Read/write la división puede equilibrar eficazmente las cargas en todo el sistema.

La coherencia de los datos requiere una organización cuidadosa en todos los servicios distribuidos. Los equipos deben implementar posibles patrones de coherencia cuando proceda y establecer límites claros de propiedad de los datos. Una supervisión sólida promueve la integridad de los datos en todos los servicios.

Además, la seguridad debe evolucionar para adaptarse a la arquitectura distribuida. Cada servicio necesita controles de seguridad detallados y sus patrones de acceso requieren una revisión periódica. La mejora de la supervisión y la auditoría se vuelven fundamentales en este entorno distribuido.

Cambios organizativos para dar soporte a las arquitecturas distribuidas

La estructura del equipo debe ajustarse a los límites de los servicios para definir claramente la propiedad y la responsabilidad. Las organizaciones deben establecer nuevos patrones de comunicación y desarrollar capacidades técnicas adicionales dentro de los equipos. Esta estructura debería respaldar tanto el mantenimiento de los servicios existentes como la evolución continua de la arquitectura.

Debe actualizar sus procesos operativos para gestionar la arquitectura distribuida. Los equipos deben modificar los procedimientos de despliegue, adaptar los procesos de respuesta a incidentes y desarrollar las prácticas de gestión de cambios para coordinar varios servicios.

Recursos

Los siguientes recursos y herramientas adicionales pueden ayudar a su organización en su proceso de descomposición de bases de datos.

AWS Guía prescriptiva

- Migración de Oracle bases de datos a Nube de AWS
- Opciones de replataforma para uno Oracle DatabaseAWS
- Patrones, arquitecturas e implementaciones de diseño en la nube

AWS publicaciones de blog

- Migre la lógica empresarial de la base de datos a la aplicación para acelerar la innovación y la flexibilidad

Servicios de AWS

- AWS Application Migration Service
- AWS Database Migration Service (AWS DMS)
- Migration Evaluator
- AWS Schema Conversion Tool (AWS SCT)
- AWS Transform

Otras herramientas

- AppEngine (sitio web de Dynatrace)
- Oracle Automatic Workload Repository (sitio web de Oracle)
- CAST Imaging (sitio web de CAST)
- Kiro (sitio web de Kiro)
- pgAdmin (sitio web de pgAdmin)
- pg_stat_statements (sitio web de PostgreSQL)

- SchemaSpy (sitio web de SchemaSpy)
- SQL Developer (sitio web de Oracle)
- SQLWays (sitio web de Ispirer)
- vFunction (sitio web de vFunction)

Otros recursos

- Monolith a los microservicios (sitio web) O'Reilly

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Preguntas frecuentes sobre mainframe y herramientas de IA	Añadimos la sección ¿Se aplican estas recomendaciones a las bases de datos monolíticas de mainframe? Preguntas frecuentes y añadimos información adicional sobre las herramientas de IA que puede utilizar durante la descomposición de las bases de datos.	14 de octubre de 2025
Publicación inicial	—	30 de septiembre de 2025

AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por la Guía AWS prescriptiva. Para sugerir entradas, utilice el enlace Enviar comentarios al final del glosario.

Números

Las 7 R

Siete estrategias de migración comunes para trasladar aplicaciones a la nube. Estas estrategias se basan en las 5 R que Gartner identificó en 2011 y consisten en lo siguiente:

- Refactorizar/rediseñar: traslade una aplicación y modifique su arquitectura mediante el máximo aprovechamiento de las características nativas en la nube para mejorar la agilidad, el rendimiento y la escalabilidad. Por lo general, esto implica trasladar el sistema operativo y la base de datos. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Amazon Aurora PostgreSQL-Compatible Edition.
- Redefinir la plataforma (transportar y redefinir): traslade una aplicación a la nube e introduzca algún nivel de optimización para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos Oracle en las instalaciones a Amazon Relational Database Service (Amazon RDS) para Oracle en la nube de Nube de AWS.
- Recomprar (readquirir): cambie a un producto diferente, lo cual se suele llevar a cabo al pasar de una licencia tradicional a un modelo SaaS. Ejemplo: Migrar el sistema de administración de las relaciones con los clientes (CRM) a Salesforce.com.
- Volver a alojar (migrar mediante lift-and-shift): traslade una aplicación a la nube sin realizar cambios para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Oracle en una instancia de EC2 en la Nube de AWS.
- Reubicar: (migrar el hipervisor mediante lift and shift): traslade la infraestructura a la nube sin comprar equipo nuevo, reescribir aplicaciones o modificar las operaciones actuales. Los servidores se migran de una plataforma en las instalaciones a un servicio en la nube para la misma plataforma. Ejemplo: migrar una Microsoft Hyper-V aplicación a AWS.
- Retener (revisitar): conserve las aplicaciones en el entorno de origen. Estas pueden incluir las aplicaciones que requieren una refactorización importante, que desee posponer para más adelante, y las aplicaciones heredadas que desee retener, ya que no hay ninguna justificación empresarial para migrarlas.

- Retirar: retire o elimine las aplicaciones que ya no sean necesarias en un entorno de origen.

A

ABAC

Consulte control de acceso basado en atributos.

servicios abstractos

Consulte servicios administrados.

ACID

Consulte atomicidad, consistencia, aislamiento, durabilidad.

migración activa-activa

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas (mediante una herramienta de replicación bidireccional o mediante operaciones de escritura doble) y ambas bases de datos gestionan las transacciones de las aplicaciones conectadas durante la migración. Este método permite la migración en lotes pequeños y controlados, en lugar de requerir una transición única. Es más flexible, pero requiere más trabajo que una migración activa-pasiva.

migración activa-pasiva

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas, pero solo la de origen gestiona las transacciones de las aplicaciones conectadas, mientras los datos se replican en la de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

función de agregación

Función SQL que actúa en un grupo de filas y calcula un único valor de devolución para el grupo. Entre los ejemplos de funciones de agregación se incluyen `SUM` y `MAX`.

IA

Consulte inteligencia artificial.

AIOps

Consulte operaciones de inteligencia artificial

anonimización

El proceso de eliminar permanentemente la información personal de un conjunto de datos. La anonimización puede ayudar a proteger la privacidad personal. Los datos anonimizados ya no se consideran datos personales.

antipatrones

Una solución que se utiliza con frecuencia para un problema recurrente en el que la solución es contraproducente, ineficaz o menos eficaz que una alternativa.

control de aplicaciones

Enfoque de seguridad que permite usar de manera exclusiva aplicaciones aprobadas para ayudar a proteger un sistema contra el malware.

cartera de aplicaciones

Recopilación de información detallada sobre cada aplicación que utiliza una organización, incluido el costo de creación y mantenimiento de la aplicación y su valor empresarial. Esta información es clave para el proceso de detección y análisis de la cartera y ayuda a identificar y priorizar las aplicaciones que se van a migrar, modernizar y optimizar.

inteligencia artificial (IA)

El campo de la informática que se dedica al uso de tecnologías informáticas para realizar funciones cognitivas que suelen estar asociadas a los seres humanos, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. Para más información, consulte ¿Qué es la inteligencia artificial?

operaciones de inteligencia artificial (AIOps)

El proceso de utilizar técnicas de machine learning para resolver problemas operativos, reducir los incidentes operativos y la intervención humana, y mejorar la calidad del servicio. Para obtener más información sobre cómo AIOps se utiliza en la estrategia de AWS migración, consulte la guía de integración de operaciones.

cifrado asimétrico

Algoritmo de cifrado que utiliza un par de claves, una clave pública para el cifrado y una clave privada para el descifrado. Puede compartir la clave pública porque no se utiliza para el descifrado, pero el acceso a la clave privada debe estar sumamente restringido.

atomicidad, consistencia, aislamiento, durabilidad (ACID)

Conjunto de propiedades de software que garantizan la validez de los datos y la fiabilidad operativa de una base de datos, incluso en caso de errores, cortes de energía u otros problemas.

control de acceso basado en atributos (ABAC)

La práctica de crear permisos detallados basados en los atributos del usuario, como el departamento, el puesto de trabajo y el nombre del equipo. Para obtener más información, consulte ABAC AWS en la documentación AWS Identity and Access Management (IAM).

origen de datos fidedigno

Ubicación en la que se almacena la versión principal de los datos, que se considera la fuente de información más fiable. Puede copiar los datos del origen de datos autorizado a otras ubicaciones con el fin de procesarlos o modificarlos, por ejemplo, anonimizarlos, redactarlos o seudonimizarlos.

Zona de disponibilidad

Una ubicación distinta dentro de una Región de AWS que está aislada de los fallos en otras zonas de disponibilidad y que proporciona una conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma región.

AWS Marco de adopción de la nube (AWS CAF)

Un marco de directrices y mejores prácticas AWS para ayudar a las organizaciones a desarrollar un plan eficiente y eficaz para migrar con éxito a la nube. AWS CAF organiza la orientación en seis áreas de enfoque denominadas perspectivas: negocios, personas, gobierno, plataforma, seguridad y operaciones. Las perspectivas empresariales, humanas y de gobernanza se centran en las habilidades y los procesos empresariales; las perspectivas de plataforma, seguridad y operaciones se centran en las habilidades y los procesos técnicos. Por ejemplo, la perspectiva humana se dirige a las partes interesadas que se ocupan de los Recursos Humanos (RR. HH.), las funciones del personal y la administración de las personas. Desde esta perspectiva, AWS CAF proporciona orientación para el desarrollo, la formación y la comunicación de las personas a fin de preparar a la organización para una adopción exitosa de la nube. Para obtener más información, consulte la Página web de AWS CAF y el Documento técnico de AWS CAF.

AWS Marco de calificación de la carga de trabajo (AWS WQF)

Herramienta que evalúa las cargas de trabajo de migración de bases de datos, recomienda estrategias de migración y proporciona estimaciones de trabajo. AWS WQF se incluye con AWS

Schema Conversion Tool ().AWS SCT Analiza los esquemas de bases de datos y los objetos de código, el código de las aplicaciones, las dependencias y las características de rendimiento y proporciona informes de evaluación.

B

bot malicioso

Bot destinado a causar interrupciones o daños a personas u organizaciones.

BCP

Consulte planificación de la continuidad del negocio.

gráfico de comportamiento

Una vista unificada e interactiva del comportamiento de los recursos y de las interacciones a lo largo del tiempo. Puede utilizar un gráfico de comportamiento con Amazon Detective para examinar los intentos de inicio de sesión fallidos, las llamadas sospechosas a la API y acciones similares. Para obtener más información, consulte Datos en un gráfico de comportamiento en la documentación de Detective.

sistema big-endian

Un sistema que almacena primero el byte más significativo. Consulte también endianidad.

clasificación binaria

Un proceso que predice un resultado binario (una de las dos clases posibles). Por ejemplo, es posible que su modelo de ML necesite predecir problemas como “¿Este correo electrónico es spam o no es spam?” o “¿Este producto es un libro o un automóvil?”.

filtro de floración

Estructura de datos probabilística y eficiente en términos de memoria que se utiliza para comprobar si un elemento es miembro de un conjunto.

implementación azul/verde

Estrategia de implementación en la que se crean dos entornos separados, pero idénticos. La versión actual de la aplicación se ejecuta en un entorno (azul) y la nueva versión de la aplicación se ejecuta en el otro entorno (verde). Esta estrategia lo ayuda a hacer reversiones rápidas con un impacto mínimo.

bot

Aplicación de software que ejecuta tareas automatizadas a través de Internet y simula la actividad o interacción humana. Algunos bots son útiles o beneficiosos, como los rastreadores web que indexan la información de Internet. Otros bots, conocidos como bots maliciosos, tienen como objetivo causar interrupciones o daños a personas u organizaciones.

botnet

Redes de bots infectadas por malware y que están bajo el control de una sola parte, conocida como pastor de bots u operador de bots. Las botnets son el mecanismo más conocido para escalar los bots y su impacto.

branch

Área contenida de un repositorio de código. La primera rama que se crea en un repositorio es la rama principal. Puede crear una rama nueva a partir de una rama existente y, a continuación, desarrollar características o corregir errores en la rama nueva. Una rama que se genera para crear una característica se denomina comúnmente rama de característica. Cuando la característica se encuentra lista para su lanzamiento, se vuelve a combinar la rama de característica con la rama principal. Para obtener más información, consulte Acerca de las sucursales (GitHub documentación).

acceso de emergencia

En circunstancias excepcionales y mediante un proceso aprobado, es una forma rápida de que un usuario pueda acceder a un Cuenta de AWS sitio al que normalmente no tiene permisos de acceso. Para más información, consulte el indicador Implement break-glass procedures en la guía de AWS Well-Architected.

estrategia de implementación sobre infraestructura existente

La infraestructura existente en su entorno. Al adoptar una estrategia de implementación sobre infraestructura existente para una arquitectura de sistemas, se diseña la arquitectura en función de las limitaciones de los sistemas y la infraestructura actuales. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

caché de búfer

El área de memoria donde se almacenan los datos a los que se accede con más frecuencia.

capacidad empresarial

Lo que hace una empresa para generar valor (por ejemplo, ventas, servicio al cliente o marketing). Las arquitecturas de microservicios y las decisiones de desarrollo pueden estar impulsadas por las capacidades empresariales. Para obtener más información, consulte la sección Organizado en torno a las capacidades empresariales del documento técnico Ejecutar microservicios en contenedores en AWS.

planificación de la continuidad del negocio (BCP)

Plan que aborda el posible impacto de un evento disruptivo, como una migración a gran escala en las operaciones y permite a la empresa reanudar las operaciones rápidamente.

C

CAF

Consulte AWS Cloud Adoption Framework.

implementación canario

Lanzamiento lento e incremental de una versión para los usuarios finales. Cuando tenga mayor confianza en la nueva versión, la implementa y reemplaza la versión actual en su totalidad.

CCoE

Consulte Centro de excelencia en la nube.

CDC

Consulte captura de datos de cambios.

captura de datos de cambio (CDC)

Proceso de seguimiento de los cambios en un origen de datos, como una tabla de base de datos, y registro de los metadatos relacionados con el cambio. Puede utilizar los CDC para diversos fines, como auditar o replicar los cambios en un sistema de destino para mantener la sincronización.

ingeniería del caos

Introducción intencionada de fallos o eventos disruptivos para poner a prueba la resiliencia de un sistema. Puedes usar AWS Fault Injection Service (AWS FIS) para realizar experimentos que estresen tus AWS cargas de trabajo y evalúen su respuesta.

CI/CD

Consulte integración continua y entrega continua.

clasificación

Un proceso de categorización que permite generar predicciones. Los modelos de ML para problemas de clasificación predicen un valor discreto. Los valores discretos siempre son distintos entre sí. Por ejemplo, es posible que un modelo necesite evaluar si hay o no un automóvil en una imagen.

cifrado del cliente

Cifrado de datos localmente, antes de que el objetivo los Servicio de AWS reciba.

Centro de excelencia en la nube (CCoE)

Equipo multidisciplinario que impulsa los esfuerzos de adopción de la nube en toda la organización, incluido el desarrollo de las prácticas recomendadas en la nube, la movilización de recursos, el establecimiento de plazos de migración y la dirección de la organización durante las transformaciones a gran escala. Para obtener más información, consulte las publicaciones de CCo E en el blog de estrategia Nube de AWS empresarial.

computación en la nube

La tecnología en la nube que se utiliza normalmente para la administración de dispositivos de IoT y el almacenamiento de datos de forma remota. La computación en la nube suele estar relacionada con la tecnología de computación de periferia.

modelo operativo en la nube

En una organización de TI, el modelo operativo que se utiliza para crear, madurar y optimizar uno o más entornos de nube. Para obtener más información, consulte Creación de su modelo operativo de nube.

etapas de adopción de la nube

Las siguientes son las cuatro fases por las que suelen pasar las empresas cuando migran a la Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realizar inversiones fundamentales para escalar su adopción de la nube (p. ej., crear una landing zone, definir una CCo E, establecer un modelo de operaciones)

- Migración: migración de aplicaciones individuales
- Reinvención: optimización de productos y servicios e innovación en la nube

Stephen Orban definió estas etapas en la entrada del blog The Journey Toward Cloud-First & the Stages of Adoption en el blog Nube de AWS Enterprise Strategy. Para obtener información sobre su relación con la estrategia de AWS migración, consulte la guía de preparación para la migración.

CMDB

Consulte base de datos de administración de configuración.

repositorio de código

Una ubicación donde el código fuente y otros activos, como documentación, muestras y scripts, se almacenan y actualizan mediante procesos de control de versiones. Algunos repositorios en la nube comunes son GitHub o Bitbucket Cloud. Cada versión del código se denomina rama. En una estructura de microservicios, cada repositorio se encuentra dedicado a una única funcionalidad. Una sola canalización de CI/CD puede utilizar varios repositorios.

caché en frío

Una caché de búfer que está vacía no está bien poblada o contiene datos obsoletos o irrelevantes. Esto afecta al rendimiento, ya que la instancia de la base de datos debe leer desde la memoria principal o el disco, lo que es más lento que leer desde la memoria caché del búfer.

datos fríos

Datos a los que se accede con poca frecuencia y que suelen ser históricos. Al consultar este tipo de datos, normalmente se aceptan consultas lentas. Trasladar estos datos a niveles o clases de almacenamiento de menor rendimiento y menos costosos puede reducir los costos.

visión artificial (CV)

Campo de la IA que utiliza el machine learning para analizar y extraer información de formatos visuales, como imágenes y videos digitales. Por ejemplo, Amazon SageMaker AI proporciona algoritmos de procesamiento de imágenes para CV.

deriva de configuración

En el caso de una carga de trabajo, un cambio en la configuración con respecto al estado esperado. Podría provocar que la carga de trabajo deje de cumplir las normas y, por lo general, es gradual e involuntaria.

base de datos de administración de configuración (CMDB)

Repositorio que almacena y administra información sobre una base de datos y su entorno de TI, incluidos los componentes de hardware y software y sus configuraciones. Por lo general, los datos de una CMDB se utilizan en la etapa de detección y análisis de la cartera de productos durante la migración.

paquete de conformidad

Un conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus controles de conformidad y seguridad. Puede implementar un paquete de conformidad como una entidad única en una región Cuenta de AWS y, o en una organización, mediante una plantilla YAML. Para obtener más información, consulta los paquetes de conformidad en la documentación. AWS Config

integración y entrega continuas (CI/CD)

El proceso de automatización de las etapas de origen, compilación, prueba, puesta en escena y producción del proceso de publicación del software. CI/CD se describe comúnmente como una canalización. CI/CD puede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar la calidad del código y entregar más rápido. Para obtener más información, consulte Beneficios de la entrega continua. CD también puede significar implementación continua. Para obtener más información, consulte Entrega continua frente a implementación continua.

CV

Consulte visión artificial.

D

datos en reposo

Datos que están estacionarios en la red, como los datos que se encuentran almacenados.

clasificación de datos

Un proceso para identificar y clasificar los datos de su red en función de su importancia y sensibilidad. Es un componente fundamental de cualquier estrategia de administración de riesgos de ciberseguridad porque lo ayuda a determinar los controles de protección y retención adecuados para los datos. La clasificación de datos es un componente del pilar de seguridad del AWS Well-Architected Framework. Para obtener más información, consulte Clasificación de datos.

deriva de datos

Una variación significativa entre los datos de producción y los datos que se utilizaron para entrenar un modelo de machine learning, o un cambio significativo en los datos de entrada a lo largo del tiempo. La deriva de datos puede reducir la calidad, la precisión y la imparcialidad generales de las predicciones de los modelos de machine learning.

datos en tránsito

Datos que se mueven de forma activa por la red, por ejemplo, entre los recursos de la red.

malla de datos

Marco de arquitectura que proporciona una propiedad de datos distribuida y descentralizada con una administración y una gobernanza centralizadas.

minimización de datos

El principio de recopilar y procesar solo los datos estrictamente necesarios. Practicar la minimización de los datos Nube de AWS puede reducir los riesgos de privacidad, los costos y la huella de carbono de la analítica.

perímetro de datos

Un conjunto de barreras preventivas en su AWS entorno que ayudan a garantizar que solo las identidades confiables accedan a los recursos confiables desde las redes esperadas. Para obtener más información, consulte Crear un perímetro de datos sobre. AWS

preprocesamiento de datos

Transformar los datos sin procesar en un formato que su modelo de ML pueda analizar fácilmente. El preprocesamiento de datos puede implicar eliminar determinadas columnas o filas y corregir los valores faltantes, incoherentes o duplicados.

procedencia de los datos

El proceso de rastrear el origen y el historial de los datos a lo largo de su ciclo de vida, por ejemplo, la forma en que se generaron, transmitieron y almacenaron los datos.

titular de los datos

Persona cuyos datos se recopilan y procesan.

almacenamiento de datos

Sistema de administración de datos que respalda la inteligencia empresarial, como los análisis. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para las consultas y los análisis.

lenguaje de definición de datos (DDL)

Instrucciones o comandos para crear o modificar la estructura de tablas y objetos de una base de datos.

lenguaje de manipulación de datos (DML)

Instrucciones o comandos para modificar (insertar, actualizar y eliminar) la información de una base de datos.

DDL

Consulte lenguaje de definición de bases de datos.

conjunto profundo

Combinar varios modelos de aprendizaje profundo para la predicción. Puede utilizar conjuntos profundos para obtener una predicción más precisa o para estimar la incertidumbre de las predicciones.

aprendizaje profundo

Un subcampo del ML que utiliza múltiples capas de redes neuronales artificiales para identificar el mapeo entre los datos de entrada y las variables objetivo de interés.

defense-in-depth

Un enfoque de seguridad de la información en el que se distribuyen cuidadosamente una serie de mecanismos y controles de seguridad en una red informática para proteger la confidencialidad, la integridad y la disponibilidad de la red y de los datos que contiene. Al adoptar esta estrategia AWS, se añaden varios controles en diferentes capas de la AWS Organizations estructura para ayudar a proteger los recursos. Por ejemplo, un defense-in-depth enfoque podría combinar la autenticación multifactorial, la segmentación de la red y el cifrado.

administrador delegado

En AWS Organizations, un servicio compatible puede registrar una cuenta de AWS miembro para administrar las cuentas de la organización y gestionar los permisos de ese servicio. Esta

cuenta se denomina administrador delegado para ese servicio. Para obtener más información y una lista de servicios compatibles, consulte Servicios que funcionan con AWS Organizations en la documentación de AWS Organizations .

Implementación

El proceso de hacer que una aplicación, características nuevas o correcciones de código se encuentren disponibles en el entorno de destino. La implementación abarca implementar cambios en una base de código y, a continuación, crear y ejecutar esa base en los entornos de la aplicación.

entorno de desarrollo

Consulte entorno.

control de detección

Un control de seguridad que se ha diseñado para detectar, registrar y alertar después de que se produzca un evento. Estos controles son una segunda línea de defensa, ya que lo advierten sobre los eventos de seguridad que han eludido los controles preventivos establecidos. Para obtener más información, consulte Controles de detección en Implementación de controles de seguridad en AWS.

asignación de flujos de valor para el desarrollo (DVSM)

Proceso que se utiliza para identificar y priorizar las restricciones que afectan negativamente a la velocidad y la calidad en el ciclo de vida del desarrollo de software. DVSM amplía el proceso de asignación del flujo de valor diseñado originalmente para las prácticas de fabricación ajustada. Se centra en los pasos y los equipos necesarios para crear y transferir valor a través del proceso de desarrollo de software.

gemelo digital

Representación virtual de un sistema del mundo real, como un edificio, una fábrica, un equipo industrial o una línea de producción. Los gemelos digitales son compatibles con el mantenimiento predictivo, la supervisión remota y la optimización de la producción.

tabla de dimensiones

En un esquema en estrella, tabla más pequeña que contiene los atributos de datos sobre los datos cuantitativos en una tabla de hechos. Los atributos de la tabla de dimensiones suelen ser campos de texto o números discretos que se comportan como texto. Estos atributos se suelen utilizar para restringir consultas, filtrarlas y etiquetar los conjuntos de resultados.

desastre

Un evento que impide que una carga de trabajo o un sistema cumplan sus objetivos empresariales en su ubicación principal de implementación. Estos eventos pueden ser desastres naturales, fallos técnicos o el resultado de acciones humanas, como una configuración incorrecta involuntaria o un ataque de malware.

recuperación de desastres (DR)

Estrategia y proceso que utiliza para minimizar el tiempo de inactividad y la pérdida de datos a causa de un desastre. Para obtener más información, consulte Recuperación ante desastres de cargas de trabajo en AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Consulte lenguaje de manipulación de bases de datos.

diseño basado en el dominio

Un enfoque para desarrollar un sistema de software complejo mediante la conexión de sus componentes a dominios en evolución, o a los objetivos empresariales principales, a los que sirve cada componente. Este concepto lo introdujo Eric Evans en su libro, Diseño impulsado por el dominio: abordando la complejidad en el corazón del software (Boston: Addison-Wesley Professional, 2003). Para obtener información sobre cómo utilizar el diseño basado en dominios con el patrón de higos estranguladores, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

DR

Consulte recuperación ante desastres.

Detección de desviaciones

Seguimiento de las desviaciones con respecto a una configuración con línea de base. Por ejemplo, puedes usarlo AWS CloudFormation para detectar desviaciones en los recursos del sistema o puedes usarlo AWS Control Tower para detectar cambios en tu landing zone que puedan afectar al cumplimiento de los requisitos de gobierno.

DVSM

Consulte asignación de flujos de valor para el desarrollo.

E

EDA

Consulte análisis de datos de tipo exploratorio.

EDI

Consulte intercambio electrónico de datos.

computación en la periferia

La tecnología que aumenta la potencia de cálculo de los dispositivos inteligentes en la periferia de una red de IoT. En comparación con la computación en la nube, la computación de periferia puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

intercambio electrónico de datos (EDI)

Intercambio automatizado de documentos comerciales entre organizaciones. Para más información, consulte ¿Qué es el intercambio electrónico de datos?

cifrado

Proceso de computación que transforma datos de texto plano, que son legibles por humanos, en texto cifrado.

clave de cifrado

Cadena criptográfica de bits aleatorios que se genera mediante un algoritmo de cifrado. Las claves pueden variar en longitud y cada una se ha diseñado para ser impredecible y única.

endianidad

El orden en el que se almacenan los bytes en la memoria del ordenador. Los sistemas big-endianos almacenan primero el byte más significativo. Los sistemas Little-Endian almacenan primero el byte menos significativo.

punto de conexión

Consulte punto de conexión de servicio.

servicio de punto de conexión

Servicio que puede alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a otros Cuentas de AWS o a responsables AWS Identity and Access Management (de IAM). Estas cuentas o

entidades principales pueden conectarse a su servicio de punto de conexión de forma privada mediante la creación de puntos de conexión de VPC de interfaz. Para obtener más información, consulte Creación de un servicio de punto de conexión en la documentación de Amazon Virtual Private Cloud (Amazon VPC).

planificación de recursos empresariales (ERP)

Sistema que automatiza y administra los procesos empresariales clave (como la contabilidad, MES y la administración de proyectos) de una empresa.

cifrado de sobre

El proceso de cifrar una clave de cifrado con otra clave de cifrado. Para obtener más información, consulte el cifrado de sobres en la documentación de AWS Key Management Service (AWS KMS).

entorno

Una instancia de una aplicación en ejecución. Los siguientes son los tipos de entornos más comunes en la computación en la nube:

- entorno de desarrollo: instancia de una aplicación en ejecución que solo se encuentra disponible para el equipo principal responsable del mantenimiento de la aplicación. Los entornos de desarrollo se utilizan para probar los cambios antes de promocionarlos a los entornos superiores. Este tipo de entorno a veces se denomina entorno de prueba.
- entornos inferiores: todos los entornos de desarrollo de una aplicación, como los que se utilizan para las compilaciones y pruebas iniciales.
- entorno de producción: instancia de una aplicación en ejecución a la que pueden acceder los usuarios finales. En un CI/CD proceso, el entorno de producción es el último entorno de implementación.
- entornos superiores: todos los entornos a los que pueden acceder usuarios que no sean del equipo de desarrollo principal. Esto puede incluir un entorno de producción, entornos de preproducción y entornos para las pruebas de aceptación por parte de los usuarios.

epopeya

En las metodologías ágiles, son categorías funcionales que ayudan a organizar y priorizar el trabajo. Las epopeyas brindan una descripción detallada de los requisitos y las tareas de implementación. Por ejemplo, las epopeyas AWS de seguridad de CAF incluyen la gestión de identidades y accesos, los controles de detección, la seguridad de la infraestructura, la protección de datos y la respuesta a incidentes. Para obtener más información sobre las epopeyas en la estrategia de migración de AWS , consulte la Guía de implementación del programa.

ERP

Consulte planificación de recursos empresariales.

análisis de datos de tipo exploratorio (EDA)

El proceso de analizar un conjunto de datos para comprender sus características principales. Se recopilan o agregan datos y, a continuación, se realizan las investigaciones iniciales para encontrar patrones, detectar anomalías y comprobar las suposiciones. El EDA se realiza mediante el cálculo de estadísticas resumidas y la creación de visualizaciones de datos.

F

tabla de hechos

Tabla central de un esquema en estrella. Almacena datos cuantitativos sobre operaciones empresariales. Por lo general, una tabla de hechos contiene dos tipos de columnas: las que contienen medidas y las que contienen una clave externa para una tabla de dimensiones.

Fail Fast

Filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de los enfoques ágiles.

límite de aislamiento de errores

En el Nube de AWS, un límite, como una zona de disponibilidad Región de AWS, un plano de control o un plano de datos, que limita el efecto de una falla y ayuda a mejorar la resiliencia de las cargas de trabajo. Para más información, consulte AWS Fault Isolation Boundaries.

rama de característica

Consulte rama.

características

Los datos de entrada que se utilizan para hacer una predicción. Por ejemplo, en un contexto de fabricación, las características pueden ser imágenes que se capturan periódicamente desde la línea de fabricación.

importancia de las características

La importancia que tiene una característica para las predicciones de un modelo. Por lo general, esto se expresa como una puntuación numérica que se puede calcular mediante diversas

técnicas, como las explicaciones aditivas de Shapley (SHAP) y los gradientes integrados. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

transformación de funciones

Optimizar los datos para el proceso de ML, lo que incluye enriquecer los datos con fuentes adicionales, escalar los valores o extraer varios conjuntos de información de un solo campo de datos. Esto permite que el modelo de ML se beneficie de los datos. Por ejemplo, si divide la fecha del “27 de mayo de 2021 00:15:37” en “jueves”, “mayo”, “2021” y “15”, puede ayudar al algoritmo de aprendizaje a aprender patrones matizados asociados a los diferentes componentes de los datos.

peticiones con pocos pasos

Proporcionar a un LLM una pequeña cantidad de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que lleve a cabo una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, mediante el que los modelos aprenden a partir de ejemplos (pasos) incrustados en las peticiones. La técnica de peticiones con pocos pasos puede ser eficaz para las tareas que requieren un formato, un razonamiento o un conocimiento del dominio específicos. Consulte también peticiones desde cero.

FGAC

Consulte control de acceso detallado.

control de acceso preciso (FGAC)

El uso de varias condiciones que tienen por objetivo permitir o denegar una solicitud de acceso.

migración relámpago

Método de migración de bases de datos que utiliza la replicación continua de datos mediante la captura de datos de cambio para migrar los datos en el menor tiempo posible, en lugar de utilizar un enfoque gradual. El objetivo es reducir al mínimo el tiempo de inactividad.

FM

Consulte modelo fundacional.

Modelo fundacional (FM)

Una gran red neuronal de aprendizaje profundo que se ha estado entrenando con conjuntos de datos masivos de datos generalizados y sin etiquetar. FMs son capaces de realizar una amplia variedad de tareas generales, como comprender el lenguaje, generar texto e imágenes

y conversar en lenguaje natural. Para más información, consulte ¿Qué son los modelos fundacionales?

G

IA generativa

Subconjunto de modelos de IA que se entrenaron con grandes cantidades de datos y que pueden utilizar una simple petición de texto para crear contenido y artefactos nuevos, como imágenes, videos, texto y audio. Para más información, consulte ¿Qué es la IA generativa?

bloqueo geográfico

Consulte restricciones geográficas.

restricciones geográficas (bloqueo geográfico)

En Amazon CloudFront, una opción para impedir que los usuarios de países específicos accedan a las distribuciones de contenido. Puede utilizar una lista de permitidos o bloqueados para especificar los países aprobados y prohibidos. Para obtener más información, consulta la sección Restringir la distribución geográfica del contenido en la CloudFront documentación.

Flujo de trabajo de Gitflow

Un enfoque en el que los entornos inferiores y superiores utilizan diferentes ramas en un repositorio de código fuente. El flujo de trabajo de Gitflow se considera heredado, mientras que el flujo de trabajo basado en enlaces troncales es el enfoque moderno preferido.

imagen dorada

Instantánea de un sistema o software que se usa como plantilla para implementar nuevas instancias de ese sistema o software. Por ejemplo, en la fabricación, una imagen dorada se puede utilizar para aprovisionar software en varios dispositivos y ayuda a mejorar la velocidad, la escalabilidad y la productividad de las operaciones de fabricación de dispositivos.

estrategia de implementación desde cero

La ausencia de infraestructura existente en un entorno nuevo. Al adoptar una estrategia de implementación desde cero para una arquitectura de sistemas, puede seleccionar todas las tecnologías nuevas sin que estas deban ser compatibles con una infraestructura existente, lo que también se conoce como implementación sobre infraestructura existente. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

barrera de protección

Una regla de alto nivel que ayuda a regular los recursos, las políticas y el cumplimiento en todas las unidades organizativas (OUs). Las barreras de protección preventivas aplican políticas para garantizar la alineación con los estándares de conformidad. Se implementan mediante políticas de control de servicios y límites de permisos de IAM. Las barreras de protección de detección detectan las vulneraciones de las políticas y los problemas de conformidad, y generan alertas para su corrección. Se implementan mediante Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

H

HA

Consulte alta disponibilidad.

migración heterogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que utilice un motor de base de datos diferente (por ejemplo, de Oracle a Amazon Aurora). La migración heterogénea suele ser parte de un esfuerzo de rediseño de la arquitectura y convertir el esquema puede ser una tarea compleja. AWS ofrece AWS SCT, lo cual ayuda con las conversiones de esquemas.

alta disponibilidad (HA)

La capacidad de una carga de trabajo para funcionar de forma continua, sin intervención, en caso de desafíos o desastres. Los sistemas de alta disponibilidad están diseñados para realizar una conmutación por error automática, ofrecer un rendimiento de alta calidad de forma constante y gestionar diferentes cargas y fallos con un impacto mínimo en el rendimiento.

modernización histórica

Un enfoque utilizado para modernizar y actualizar los sistemas de tecnología operativa (TO) a fin de satisfacer mejor las necesidades de la industria manufacturera. Un histórico es un tipo de base de datos que se utiliza para recopilar y almacenar datos de diversas fuentes en una fábrica.

datos de reserva

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de machine learning. Puede utilizar los datos de reserva para evaluar el rendimiento del modelo mediante la comparación de las predicciones del modelo con los datos de reserva.

migración homogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que comparte el mismo motor de base de datos (por ejemplo, Microsoft SQL Server a Amazon RDS para SQL Server). La migración homogénea suele formar parte de un esfuerzo para volver a alojar o redefinir la plataforma. Puede utilizar las utilidades de bases de datos nativas para migrar el esquema.

datos recientes

Datos a los que se accede con frecuencia, como datos en tiempo real o datos traslacionales recientes. Por lo general, estos datos requieren un nivel o una clase de almacenamiento de alto rendimiento para proporcionar respuestas rápidas a las consultas.

hotfix

Una solución urgente para un problema crítico en un entorno de producción. Debido a su urgencia, una revisión suele realizarse fuera del flujo de trabajo de DevOps publicación típico.

periodo de hiperatención

Periodo, inmediatamente después de la transición, durante el cual un equipo de migración administra y monitorea las aplicaciones migradas en la nube para solucionar cualquier problema. Por lo general, este periodo dura de 1 a 4 días. Al final del periodo de hiperatención, el equipo de migración suele transferir la responsabilidad de las aplicaciones al equipo de operaciones en la nube.

I

IaC

Consulte infraestructura como código.

políticas basadas en identidades

Política asociada a uno o más directores de IAM que define sus permisos en el entorno. Nube de AWS

aplicación inactiva

Aplicación que utiliza un promedio de CPU y memoria de entre 5 y 20 por ciento durante un periodo de 90 días. En un proyecto de migración, es habitual retirar estas aplicaciones o mantenerlas en las instalaciones.

IIoT

Consulte Internet de las cosas industrial.

infraestructura inmutable

Modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar o modificar la infraestructura existente o aplicarle revisiones. Las infraestructuras inmutables son de manera intrínseca más coherentes, fiables y predecibles que las infraestructuras mutables. Para más información, consulte la práctica recomendada Implementación mediante una infraestructura inmutable en el Marco de AWS Well-Architected.

VPC entrante (de entrada)

En una arquitectura de AWS cuentas múltiples, una VPC que acepta, inspecciona y enruta las conexiones de red desde fuera de una aplicación. La arquitectura AWS de referencia de seguridad recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación y el resto de Internet.

migración gradual

Estrategia de transición en la que se migra la aplicación en partes pequeñas en lugar de realizar una transición única y completa. Por ejemplo, puede trasladar inicialmente solo unos pocos microservicios o usuarios al nuevo sistema. Tras comprobar que todo funciona correctamente, puede trasladar microservicios o usuarios adicionales de forma gradual hasta que pueda retirar su sistema heredado. Esta estrategia reduce los riesgos asociados a las grandes migraciones.

Industria 4.0

Término que introdujo Klaus Schwab en 2016 para referirse a la modernización de los procesos de fabricación mediante los avances en la conectividad, los datos en tiempo real, la automatización, el análisis, la IA y el ML.

infraestructura

Todos los recursos y activos que se encuentran en el entorno de una aplicación.

infraestructura como código (IaC)

Proceso de aprovisionamiento y administración de la infraestructura de una aplicación mediante un conjunto de archivos de configuración. La IaC se ha diseñado para ayudarlo a centralizar la administración de la infraestructura, estandarizar los recursos y escalar con rapidez a fin de que los entornos nuevos sean repetibles, fiables y consistentes.

Internet de las cosas industrial (T) IIo

El uso de sensores y dispositivos conectados a Internet en los sectores industriales, como el productivo, el eléctrico, el automotriz, el sanitario, el de las ciencias de la vida y el de la agricultura. Para obtener más información, consulte Creación de una estrategia de transformación digital de la Internet de las cosas (IIoT) industrial.

VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red VPCs entre Internet y las redes locales (en una misma o Regiones de AWS diferente). La arquitectura AWS de referencia de seguridad recomienda configurar su cuenta de red con entrada, salida e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

Internet de las cosas (IoT)

Red de objetos físicos conectados con sensores o procesadores integrados que se comunican con otros dispositivos y sistemas a través de Internet o de una red de comunicación local. Para obtener más información, consulte ¿Qué es IoT?.

interpretabilidad

Característica de un modelo de machine learning que describe el grado en que un ser humano puede entender cómo las predicciones del modelo dependen de sus entradas. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

IoT

Consulte Internet de las cosas.

biblioteca de información de TI (ITIL)

Conjunto de prácticas recomendadas para ofrecer servicios de TI y alinearlos con los requisitos empresariales. La ITIL proporciona la base para la ITSM.

administración de servicios de TI (ITSM)

Actividades asociadas con el diseño, la implementación, la administración y el soporte de los servicios de TI para una organización. Para obtener información sobre la integración de las operaciones en la nube con las herramientas de ITSM, consulte la Guía de integración de operaciones.

ITIL

Consulte biblioteca de información de TI.

ITSM

Consulte administración de servicios de TI.

L

control de acceso basado en etiquetas (LBAC)

Una implementación del control de acceso obligatorio (MAC) en la que a los usuarios y a los propios datos se les asigna explícitamente un valor de etiqueta de seguridad. La intersección entre la etiqueta de seguridad del usuario y la etiqueta de seguridad de los datos determina qué filas y columnas puede ver el usuario.

zona de aterrizaje

Una landing zone es un AWS entorno multicuenta bien diseñado, escalable y seguro. Este es un punto de partida desde el cual las empresas pueden lanzar e implementar rápidamente cargas de trabajo y aplicaciones con confianza en su entorno de seguridad e infraestructura. Para obtener más información sobre las zonas de aterrizaje, consulte Configuración de un entorno de AWS seguro y escalable con varias cuentas.

modelo de lenguaje de gran tamaño (LLM)

Modelo de IA de aprendizaje profundo que se entrenó previamente con una gran cantidad de datos. Un LLM puede llevar a cabo varias tareas, como responder preguntas, resumir documentos, traducir textos a otros idiomas y completar oraciones. Para obtener más información, consulte Qué son. LLMs

migración grande

Migración de 300 servidores o más.

LBAC

Consulte control de acceso basado en etiquetas.

privilegio mínimo

La práctica recomendada de seguridad que consiste en conceder los permisos mínimos necesarios para realizar una tarea. Para obtener más información, consulte Aplicar permisos de privilegio mínimo en la documentación de IAM.

migrar mediante lift-and-shift

Consulte Las 7 R.

sistema little-endian

Un sistema que almacena primero el byte menos significativo. Consulte también endianidad.

LLM

Consulte modelo de lenguaje de gran tamaño.

entornos inferiores

Consulte entorno.

M

machine learning (ML)

Un tipo de inteligencia artificial que utiliza algoritmos y técnicas para el reconocimiento y el aprendizaje de patrones. El ML analiza y aprende de los datos registrados, como los datos del Internet de las cosas (IoT), para generar un modelo estadístico basado en patrones. Para más información, consulte Machine learning.

rama principal

Consulte rama.

malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware podría interrumpir los sistemas informáticos, filtrar información confidencial u obtener acceso no autorizado. Algunos ejemplos de malware son los virus, los gusanos, el ransomware, los troyanos, el spyware y los registradores de pulsaciones de teclas.

Servicios administrados

Servicios de AWS para lo cual AWS opera la capa de infraestructura, el sistema operativo y las plataformas, y se accede a los puntos finales para almacenar y recuperar datos. Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB son ejemplos de servicios administrados. También se conocen como servicios abstractos.

sistema de ejecución de fabricación (MES)

Sistema de software para seguir, supervisar, documentar y controlar los procesos de producción que convierten las materias primas en productos acabados en la zona de producción.

MAP

Consulte Programa de aceleración de la migración.

mecanismo

Proceso completo mediante el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para hacer ajustes. Un mecanismo es un ciclo que se refuerza y mejora por sí mismo a medida que funciona. Para obtener más información, consulte Creación de mecanismos en el AWS Well-Architected Framework.

cuenta de miembro

Todas las Cuentas de AWS demás cuentas, excepto la de administración, que forman parte de una organización. AWS Organizations Una cuenta no puede pertenecer a más de una organización a la vez.

MES

Consulte sistema de ejecución de fabricación.

Message Queuing Telemetry Transport (MQTT)

Un protocolo de comunicación ligero machine-to-machine (M2M), basado en el patrón de publicación/suscripción, para dispositivos de IoT con recursos limitados.

microservicio

Un servicio pequeño e independiente que se comunica a través de una red bien definida APIs y que, por lo general, es propiedad de equipos pequeños e independientes. Por ejemplo, un sistema de seguros puede incluir microservicios que se adapten a las capacidades empresariales, como las de ventas o marketing, o a subdominios, como las de compras, reclamaciones o análisis. Los beneficios de los microservicios incluyen la agilidad, la escalabilidad flexible, la facilidad de implementación, el código reutilizable y la resiliencia. Para obtener más información, consulte Integrar microservicios mediante AWS servicios sin servidor.

arquitectura de microservicios

Un enfoque para crear una aplicación con componentes independientes que ejecutan cada proceso de la aplicación como un microservicio. Estos microservicios se comunican a través de una interfaz bien definida mediante un uso ligero. APIs Cada microservicio de esta arquitectura se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Para obtener más información, consulte Implementación de microservicios en. AWS

Programa de aceleración de la migración (MAP)

Un AWS programa que proporciona soporte de consultoría, formación y servicios para ayudar a las organizaciones a crear una base operativa sólida para migrar a la nube y para ayudar a compensar el costo inicial de las migraciones. El MAP incluye una metodología de migración para ejecutar las migraciones antiguas de forma metódica y un conjunto de herramientas para automatizar y acelerar los escenarios de migración más comunes.

migración a escala

Proceso de transferencia de la mayoría de la cartera de aplicaciones a la nube en oleadas, con más aplicaciones desplazadas a un ritmo más rápido en cada oleada. En esta fase, se utilizan las prácticas recomendadas y las lecciones aprendidas en las fases anteriores para implementar una fábrica de migración de equipos, herramientas y procesos con el fin de agilizar la migración de las cargas de trabajo mediante la automatización y la entrega ágil. Esta es la tercera fase de la estrategia de migración de AWS.

fábrica de migración

Equipos multifuncionales que agilizan la migración de las cargas de trabajo mediante enfoques automatizados y ágiles. Los equipos de las fábricas de migración suelen incluir a analistas y propietarios de operaciones, empresas, ingenieros de migración, desarrolladores y DevOps profesionales que trabajan a pasos agigantados. Entre el 20 y el 50 por ciento de la cartera de aplicaciones empresariales se compone de patrones repetidos que pueden optimizarse mediante un enfoque de fábrica. Para obtener más información, consulte la discusión sobre las fábricas de migración y la Guía de fábricas de migración a la nube en este contenido.

metadatos de migración

Información sobre la aplicación y el servidor que se necesita para completar la migración. Cada patrón de migración requiere un conjunto diferente de metadatos de migración. Algunos ejemplos de metadatos de migración son la subred de destino, el grupo de seguridad y AWS la cuenta.

patrón de migración

Tarea de migración repetible que detalla la estrategia de migración, el destino de la migración y la aplicación o el servicio de migración utilizados. Ejemplo: rehospede la migración a Amazon EC2 AWS con Application Migration Service.

Migration Portfolio Assessment (MPA)

Herramienta en línea que proporciona información a fin de validar los argumentos comerciales necesarios para migrar a la Nube de AWS. La MPA ofrece una evaluación detallada de la cartera

(adecuación del tamaño de los servidores, precios, comparaciones del costo total de propiedad, análisis de los costos de migración), así como una planificación de la migración (análisis y recopilación de datos de aplicaciones, agrupación de aplicaciones, priorización de la migración y planificación de oleadas). La herramienta MPA (requiere iniciar sesión) está disponible de forma gratuita para todos los AWS consultores y consultores de los socios de APN.

Evaluación de la preparación para la migración (MRA)

Proceso que consiste en obtener información sobre el estado de preparación de una organización para la nube, identificar sus puntos fuertes y débiles y elaborar un plan de acción para cerrar las brechas identificadas mediante el AWS CAF. Para obtener más información, consulte la Guía de preparación para la migración. La MRA es la primera fase de la estrategia de migración de AWS.

estrategia de migración

Enfoque utilizado para migrar una carga de trabajo a la Nube de AWS. Para más información, consulte la entrada Las 7 R de este glosario y también Mobilize your organization to accelerate large-scale migrations.

ML

Consulte machine learning.

modernización

Transformar una aplicación obsoleta (antigua o monolítica) y su infraestructura en un sistema ágil, elástico y de alta disponibilidad en la nube para reducir los gastos, aumentar la eficiencia y aprovechar las innovaciones. Para más información, consulte Strategy for modernizing applications in the Nube de AWS.

evaluación de la preparación para la modernización

Evaluación que ayuda a determinar la preparación para la modernización de las aplicaciones de una organización; identifica los beneficios, los riesgos y las dependencias; y determina qué tan bien la organización puede soportar el estado futuro de esas aplicaciones. El resultado de la evaluación es un esquema de la arquitectura objetivo, una hoja de ruta que detalla las fases de desarrollo y los hitos del proceso de modernización y un plan de acción para abordar las brechas identificadas. Para más información, consulte Evaluating modernization readiness for applications in the Nube de AWS.

aplicaciones monolíticas (monolitos)

Aplicaciones que se ejecutan como un único servicio con procesos estrechamente acoplados. Las aplicaciones monolíticas presentan varios inconvenientes. Si una característica de la

aplicación experimenta un aumento en la demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica también se vuelve más complejo a medida que crece la base de código. Para solucionar problemas con la aplicación, puede utilizar una arquitectura de microservicios. Para obtener más información, consulte Descomposición de monolitos en microservicios.

MPA

Consulte Migration Portfolio Assessment.

MQTT

Consulte Message Queuing Telemetry Transport.

clasificación multiclase

Un proceso que ayuda a generar predicciones para varias clases (predice uno de más de dos resultados). Por ejemplo, un modelo de ML podría preguntar "¿Este producto es un libro, un automóvil o un teléfono?" o "¿Qué categoría de productos es más interesante para este cliente?".

infraestructura mutable

Modelo que actualiza y modifica la infraestructura actual para las cargas de trabajo de producción. Para mejorar la coherencia, la fiabilidad y la previsibilidad, el AWS Well-Architected Framework recomienda el uso de una infraestructura inmutable como práctica recomendada.

O

OAC

Consulte control de acceso de origen.

OAI

Consulte identidad de acceso de origen.

OCM

Consulte administración del cambio organizacional.

migración fuera de línea

Método de migración en el que la carga de trabajo de origen se elimina durante el proceso de migración. Este método implica un tiempo de inactividad prolongado y, por lo general, se utiliza para cargas de trabajo pequeñas y no críticas.

OI

Consulte integración de operaciones.

OLA

Consulte acuerdo de nivel operativo.

migración en línea

Método de migración en el que la carga de trabajo de origen se copia al sistema de destino sin que se desconecte. Las aplicaciones que están conectadas a la carga de trabajo pueden seguir funcionando durante la migración. Este método implica un tiempo de inactividad nulo o mínimo y, por lo general, se utiliza para cargas de trabajo de producción críticas.

OPC-UA

Consulte Open Process Communications: arquitectura unificada.

Open Process Communications: arquitectura unificada (OPC-UA)

Un protocolo de machine-to-machine comunicación (M2M) para la automatización industrial. OPC-UA establece un estándar de interoperabilidad con esquemas de autenticación, autorización y cifrado de datos.

acuerdo de nivel operativo (OLA)

Acuerdo que aclara lo que los grupos de TI operativos se comprometen a ofrecerse entre sí, para respaldar un acuerdo de nivel de servicio (SLA).

revisión de la preparación operativa (ORR)

Lista de comprobación de preguntas y prácticas recomendadas asociadas que son útiles para comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles errores. Para más información, consulte Operational Readiness Reviews (ORR) en el Marco de AWS Well-Architected.

tecnología operativa (TO)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En el sector de la fabricación, la integración de los sistemas de TO y tecnología de la información (TI) es un enfoque clave para las transformaciones de la industria 4.0.

integración de operaciones (OI)

Proceso de modernización de las operaciones en la nube, que implica la planificación de la preparación, la automatización y la integración. Para obtener más información, consulte la Guía de integración de las operaciones.

registro de seguimiento organizativo

Un registro creado por y AWS CloudTrail que registra todos los eventos para todos los miembros Cuentas de AWS de una organización. AWS Organizations Este registro de seguimiento se crea en cada Cuenta de AWS que forma parte de la organización y realiza un seguimiento de la actividad en cada cuenta. Para obtener más información, consulte Crear un registro para una organización en la CloudTrail documentación.

administración del cambio organizacional (OCM)

Marco para administrar las transformaciones empresariales importantes y disruptivas desde la perspectiva de las personas, la cultura y el liderazgo. La OCM ayuda a las empresas a prepararse para nuevos sistemas y estrategias y a realizar la transición a ellos, al acelerar la adopción de cambios, abordar los problemas de transición e impulsar cambios culturales y organizacionales. En la estrategia de AWS migración, este marco se denomina aceleración de personal, debido a la velocidad de cambio que requieren los proyectos de adopción de la nube. Para obtener más información, consulte la Guía de OCM.

control de acceso de origen (OAC)

En CloudFront, una opción mejorada para restringir el acceso y proteger el contenido del Amazon Simple Storage Service (Amazon S3). El OAC admite todos los buckets de S3 Regiones de AWS, el cifrado del lado del servidor AWS KMS (SSE-KMS) y las solicitudes dinámicas `PUT` y `DELETE` dirigidas al bucket de S3.

identidad de acceso de origen (OAI)

En CloudFront, una opción para restringir el acceso y proteger el contenido de Amazon S3. Cuando utiliza OAI, CloudFront crea un principal con el que Amazon S3 puede autenticarse. Los directores autenticados solo pueden acceder al contenido de un bucket de S3 a través de una distribución específica. CloudFront Consulte también el OAC, que proporciona un control de acceso más detallado y mejorado.

ORR

Consulte revisión de la preparación operativa.

OT

Consulte tecnología operativa.

VPC saliente (de salida)

En una arquitectura de AWS cuentas múltiples, una VPC que gestiona las conexiones de red que se inician desde una aplicación. La arquitectura AWS de referencia de seguridad recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

P

límite de permisos

Una política de administración de IAM que se adjunta a las entidades principales de IAM para establecer los permisos máximos que puede tener el usuario o el rol. Para obtener más información, consulte Límites de permisos en la documentación de IAM.

información de identificación personal (PII)

Información que, vista directamente o combinada con otros datos relacionados, puede utilizarse para deducir de manera razonable la identidad de una persona. Algunos ejemplos de información de identificación personal son los nombres, las direcciones y la información de contacto.

PII

Consulte información de identificación personal.

manual de estrategias

Conjunto de pasos predefinidos que capturan el trabajo asociado a las migraciones, como la entrega de las funciones de operaciones principales en la nube. Un manual puede adoptar la forma de scripts, manuales de procedimientos automatizados o resúmenes de los procesos o pasos necesarios para operar un entorno modernizado.

PLC

Consulte controlador lógico programable.

PLM

Consulte administración del ciclo de vida del producto.

policy

Objeto que puede definir permisos (consulte política basada en identidad), especificar las condiciones de acceso (consulte política basada en recursos) o definir los permisos máximos para todas las cuentas de una organización de AWS Organizations (consulte política de control de servicio).

persistencia políglota

Elegir de forma independiente la tecnología de almacenamiento de datos de un microservicio en función de los patrones de acceso a los datos y otros requisitos. Si sus microservicios tienen la misma tecnología de almacenamiento de datos, pueden enfrentarse a desafíos de implementación o experimentar un rendimiento deficiente. Los microservicios se implementan más fácilmente y logran un mejor rendimiento y escalabilidad si utilizan el almacén de datos que mejor se adapte a sus necesidades.

evaluación de cartera

Proceso de detección, análisis y priorización de la cartera de aplicaciones para planificar la migración. Para obtener más información, consulte la Evaluación de la preparación para la migración.

predicate

Condición de consulta que devuelve `true` o `false`. En general, se encuentra en una cláusula `WHERE`.

inserción de predicados

Técnica de optimización de consultas en bases de datos que filtra los datos de la consulta antes de transferirlos. Esta técnica reduce la cantidad de datos de la base de datos relacional que se tienen que recuperar y procesar. Además, mejora el rendimiento de las consultas.

control preventivo

Un control de seguridad diseñado para evitar que ocurra un evento. Estos controles son la primera línea de defensa para evitar el acceso no autorizado o los cambios no deseados en la red. Para obtener más información, consulte Controles preventivos en Implementación de controles de seguridad en AWS.

entidad principal

Una entidad AWS que puede realizar acciones y acceder a los recursos. Esta entidad suele ser un usuario raíz para un Cuenta de AWS rol de IAM o un usuario. Para obtener más información, consulte Entidad principal en Términos y conceptos de roles en la documentación de IAM.

Privacidad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la privacidad durante todo el proceso de desarrollo.

zonas alojadas privadas

Un contenedor que contiene información sobre cómo desea que Amazon Route 53 responda a las consultas de DNS de un dominio y sus subdominios dentro de uno o más VPCs. Para obtener más información, consulte Uso de zonas alojadas privadas en la documentación de Route 53.

control proactivo

Control de seguridad que se diseñó para evitar la implementación de recursos que no cumplan con la normativa. Estos controles analizan los recursos antes de aprovisionarlos. Si el recurso no cumple con los requisitos del control, no se aprovisiona. Para obtener más información, consulte la guía de referencia de controles en la AWS Control Tower documentación y consulte Controles proactivos en la sección Implementación de controles de seguridad en AWS.

administración del ciclo de vida del producto (PLM)

Administración de los datos y los procesos de un producto a lo largo de todo su ciclo de vida, desde el diseño, el desarrollo y el lanzamiento, pasando por el crecimiento y la madurez, hasta la reducción de su uso y su retirada.

entorno de producción

Consulte entorno.

controlador lógico programable (PLC)

En el sector de la fabricación, computadora adaptable y altamente fiable que supervisa las máquinas y automatiza los procesos de fabricación.

encadenamiento de peticiones

Uso de la salida de una petición de LLM como entrada para la siguiente petición a fin de generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en tareas secundarias o para refinar o ampliar de forma iterativa una respuesta preliminar. Ayuda a mejorar la precisión y la relevancia de las respuestas de un modelo y permite obtener resultados más detallados y personalizados.

seudonimización

El proceso de reemplazar los identificadores personales de un conjunto de datos por valores de marcadores de posición. La seudonimización puede ayudar a proteger la privacidad personal. Los datos seudonimizados siguen considerándose datos personales.

publish/subscribe (pub/sub)

Patrón que permite establecer comunicaciones asíncronas entre microservicios para mejorar la escalabilidad y la capacidad de respuesta. Por ejemplo, en un MES basado en microservicios, un microservicio puede publicar mensajes de eventos en un canal al que se pueden suscribir otros microservicios. El sistema puede agregar nuevos microservicios sin cambiar el servicio de publicación.

Q

plan de consulta

Serie de pasos, como instrucciones, que se utilizan para acceder a los datos de un sistema de base de datos relacional SQL.

regresión del plan de consulta

El optimizador de servicios de la base de datos elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos. Los cambios en estadísticas, restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan.

R

Matriz RACI

Consulte responsable, fiable, consultada e informada (RACI).

RAG

Consulte generación aumentada por recuperación.

ransomware

Software malicioso que se ha diseñado para bloquear el acceso a un sistema informático o a los datos hasta que se efectúe un pago.

Matriz RASCI

Consulte responsable, fiable, consultada e informada (RACI).

RCAC

Consulte control de acceso por filas y columnas.

réplica de lectura

Una copia de una base de datos que se utiliza con fines de solo lectura. Puede enrutar las consultas a la réplica de lectura para reducir la carga en la base de datos principal.

rediseñar

Consulte Las 7 R.

objetivo de punto de recuperación (RPO)

La cantidad de tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y la interrupción del servicio.

objetivo de tiempo de recuperación (RTO)

La demora máxima aceptable entre la interrupción del servicio y el restablecimiento del servicio.

refactorizar

Consulte Las 7 R.

Region

Conjunto de AWS recursos en un área geográfica. Cada uno Región de AWS está aislado e independiente de los demás para proporcionar tolerancia a las fallas, estabilidad y resiliencia. Para más información, consulte Specify which Regiones de AWS your account can use.

regresión

Una técnica de ML que predice un valor numérico. Por ejemplo, para resolver el problema de "¿A qué precio se venderá esta casa?", un modelo de ML podría utilizar un modelo de regresión lineal para predecir el precio de venta de una vivienda en función de datos conocidos sobre ella (por ejemplo, los metros cuadrados).

volver a alojar

Consulte Las 7 R.

versión

En un proceso de implementación, el acto de promover cambios en un entorno de producción.

reubicar

Consulte Las 7 R.

redefinir la plataforma

Consulte Las 7 R.

recomprar

Consulte Las 7 R.

resiliencia

Capacidad de una aplicación para resistir interrupciones o recuperarse de ellas. Al planificar la resiliencia en la Nube de AWS, la alta disponibilidad y la recuperación ante desastres son consideraciones comunes. Para más información, consulte Resiliencia en la Nube de AWS.

política basada en recursos

Una política asociada a un recurso, como un bucket de Amazon S3, un punto de conexión o una clave de cifrado. Este tipo de política especifica a qué entidades principales se les permite el acceso, las acciones compatibles y cualquier otra condición que deba cumplirse.

matriz responsable, confiable, consultada e informada (RACI)

Una matriz que define las funciones y responsabilidades de todas las partes involucradas en las actividades de migración y las operaciones de la nube. El nombre de la matriz se deriva de los tipos de responsabilidad definidos en la matriz: responsable (R), contable (A), consultado (C) e informado (I). El tipo de soporte (S) es opcional. Si incluye el soporte, la matriz se denomina matriz RASCI y, si la excluye, se denomina matriz RACI.

control receptivo

Un control de seguridad que se ha diseñado para corregir los eventos adversos o las desviaciones con respecto a su base de seguridad. Para obtener más información, consulte Controles receptivos en Implementación de controles de seguridad en AWS.

retain

Consulte Las 7 R.

retirar

Consulte Las 7 R.

Generación aumentada de recuperación (RAG)

Tecnología de IA generativa mediante la que un LLM hace referencia a un origen de datos autorizado que se encuentra fuera de sus orígenes de datos de entrenamiento antes de generar una respuesta. Por ejemplo, un modelo de RAG podría hacer una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para más información, consulte ¿Qué es RAG (generación aumentada por recuperación)?

rotación

Proceso mediante el que periódicamente se actualiza un secreto para que resulte más difícil que un atacante pueda acceder a las credenciales.

control de acceso por filas y columnas (RCAC)

El uso de expresiones SQL básicas y flexibles que tienen reglas de acceso definidas. El RCAC consta de permisos de fila y máscaras de columnas.

RPO

Consulte objetivo de punto de recuperación.

RTO

Consulte objetivo de tiempo de recuperación.

manual de procedimientos

Conjunto de procedimientos manuales o automatizados necesarios para realizar una tarea específica. Por lo general, se diseñan para agilizar las operaciones o los procedimientos repetitivos con altas tasas de error.

S

SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión Consola de administración de AWS o llamar a las operaciones de la AWS API sin tener que crear un

usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte Acerca de la federación basada en SAML 2.0 en la documentación de IAM.

SCADA

Consulte control de supervisión y adquisición de datos.

SCP

Consulte política de control de servicio.

secreta

En AWS Secrets Manager, información confidencial o restringida, como una contraseña o credenciales de usuario, que se almacena de forma cifrada. Se compone del valor del secreto y de sus metadatos. El valor del secreto puede ser binario, una sola cadena o varias cadenas. Para más información, consulte What's in a Secrets Manager secret? en la documentación de Secrets Manager.

seguridad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la seguridad durante todo el proceso de desarrollo.

control de seguridad

Barrera de protección técnica o administrativa que impide, detecta o reduce la capacidad de un agente de amenazas para aprovechar una vulnerabilidad de seguridad. Existen cuatro tipos de controles de seguridad principales: preventivos, de detección, de respuesta y proactivos.

refuerzo de la seguridad

Proceso de reducir la superficie expuesta a ataques para hacerla más resistente a los ataques. Esto puede incluir acciones, como la eliminación de los recursos que ya no se necesitan, la implementación de prácticas recomendadas de seguridad consistente en conceder privilegios mínimos o la desactivación de características innecesarias en los archivos de configuración.

sistema de información sobre seguridad y administración de eventos (SIEM)

Herramientas y servicios que combinan sistemas de administración de información sobre seguridad (SIM) y de administración de eventos de seguridad (SEM). Un sistema de SIEM recopila, monitorea y analiza los datos de servidores, redes, dispositivos y otras fuentes para detectar amenazas y brechas de seguridad y generar alertas.

automatización de la respuesta de seguridad

Acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o corregirlo. Estas automatizaciones sirven como controles de seguridad preventivos o adaptables que le ayudan a implementar las mejores prácticas AWS de seguridad. La modificación de un grupo de seguridad de VPC, la aplicación de revisiones a una instancia de Amazon EC2 o la rotación de credenciales son algunos ejemplos de acciones de respuesta automatizadas.

cifrado del servidor

Cifrado de los datos en su destino, por parte de Servicio de AWS quien los recibe.

política de control de servicio (SCP)

Política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. SCPs defina barreras o establezca límites a las acciones que un administrador puede delegar en usuarios o roles. Puede utilizarlas SCPs como listas de permitidos o rechazados para especificar qué servicios o acciones están permitidos o prohibidos. Para obtener más información, consulte las políticas de control de servicios en la AWS Organizations documentación.

punto de enlace de servicio

La URL del punto de entrada de un Servicio de AWS. Para conectarse mediante programación a un servicio de destino, puede utilizar un punto de conexión. Para obtener más información, consulte Puntos de conexión de Servicio de AWS en Referencia general de AWS.

acuerdo de nivel de servicio (SLA)

Acuerdo que aclara lo que un equipo de TI se compromete a ofrecer a los clientes, como el tiempo de actividad y el rendimiento del servicio.

indicador de nivel de servicio (SLI)

Medición de un aspecto del rendimiento de un servicio, como la tasa de errores, la disponibilidad o el rendimiento.

objetivo de nivel de servicio (SLO)

Métrica objetivo que representa el estado de un servicio medido mediante un indicador de nivel de servicio.

modelo de responsabilidad compartida

Un modelo que describe la responsabilidad con AWS la que compartes la seguridad y el cumplimiento de la nube. AWS es responsable de la seguridad de la nube, mientras que usted es responsable de la seguridad en la nube. Para obtener más información, consulte el Modelo de responsabilidad compartida.

SIEM

Consulte sistema de administración de eventos e información de seguridad.

único punto de error (SPOF)

Error en un único componente crítico de una aplicación que puede interrumpir el sistema.

SLA

Consulte acuerdo de nivel de servicio.

SLI

Consulte indicador de nivel de servicio.

SLO

Consulte objetivo de nivel de servicio.

split-and-seed modelo

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para más información, consulte Phased approach to modernizing applications in the Nube de AWS.

SPOF

Consulte único punto de error.

esquema en estrella

Estructura organizativa de una base de datos que utiliza una tabla de hechos de gran tamaño para almacenar datos transaccionales o medidos y una o varias tablas dimensionales más pequeñas para almacenar los atributos de los datos. Esta estructura está diseñada para utilizarse en un almacén de datos o con fines de inteligencia empresarial.

patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda desmantelar el sistema heredado. Este patrón utiliza la analogía de una higuera que crece hasta convertirse en un árbol estable y, finalmente, se apodera y reemplaza a su host. El patrón fue presentado por Martin Fowler como una forma de gestionar el riesgo al reescribir sistemas monolíticos. Para ver un ejemplo con la aplicación de este patrón, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

subred

Un intervalo de direcciones IP en la VPC. Una subred debe residir en una sola zona de disponibilidad.

control de supervisión y adquisición de datos (SCADA)

En el sector de la fabricación, sistema que utiliza hardware y software para supervisar los activos físicos y las operaciones de producción.

cifrado simétrico

Un algoritmo de cifrado que utiliza la misma clave para cifrar y descifrar los datos.

pruebas sintéticas

Prueba de un sistema de manera que simule las interacciones de los usuarios para detectar posibles problemas o supervisar el rendimiento. Puede usar Amazon CloudWatch Synthetics para crear estas pruebas.

petición del sistema

Técnica para proporcionar contexto, instrucciones o pautas a un LLM para dirigir su comportamiento. Las peticiones del sistema ayudan a establecer el contexto y las reglas para las interacciones con los usuarios.

T

etiquetas

Pares clave-valor que actúan como metadatos para organizar los recursos. AWS Las etiquetas pueden ayudar a administrar, identificar, organizar, buscar y filtrar recursos de . Para obtener más información, consulte Etiquetado de los recursos de AWS.

variable de destino

El valor que intenta predecir en el ML supervisado. Esto también se conoce como variable de resultado. Por ejemplo, en un entorno de fabricación, la variable objetivo podría ser un defecto del producto.

lista de tareas

Herramienta que se utiliza para hacer un seguimiento del progreso mediante un manual de procedimientos. La lista de tareas contiene una descripción general del manual de procedimientos y una lista de las tareas generales que deben completarse. Para cada tarea general, se incluye la cantidad estimada de tiempo necesario, el propietario y el progreso.

entorno de prueba

Consulte entorno.

entrenamiento

Proporcionar datos de los que pueda aprender su modelo de ML. Los datos de entrenamiento deben contener la respuesta correcta. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir). Genera un modelo de ML que captura estos patrones. Luego, el modelo de ML se puede utilizar para obtener predicciones sobre datos nuevos para los que no se conoce el destino.

puerta de enlace de tránsito

Un centro de tránsito de red que puede usar para interconectar sus redes con VPCs las locales. Para obtener más información, consulte Qué es una pasarela de tránsito en la AWS Transit Gateway documentación.

flujo de trabajo basado en enlaces troncales

Un enfoque en el que los desarrolladores crean y prueban características de forma local en una rama de característica y, a continuación, combinan esos cambios en la rama principal. Luego, la rama principal se adapta a los entornos de desarrollo, preproducción y producción, de forma secuencial.

acceso de confianza

Otorgar permisos a un servicio que especifique para realizar tareas en su organización AWS Organizations y en sus cuentas en su nombre. El servicio de confianza crea un rol vinculado al servicio en cada cuenta, cuando ese rol es necesario, para realizar las tareas de administración

por usted. Para obtener más información, consulte AWS Organizations Utilización con otros AWS servicios en la AWS Organizations documentación.

ajuste

Cambiar aspectos de su proceso de formación a fin de mejorar la precisión del modelo de ML. Por ejemplo, puede entrenar el modelo de ML al generar un conjunto de etiquetas, incorporar etiquetas y, luego, repetir estos pasos varias veces con diferentes ajustes para optimizar el modelo.

equipo de dos pizzas

Un DevOps equipo pequeño al que puedes alimentar con dos pizzas. Un equipo formado por dos integrantes garantiza la mejor oportunidad posible de colaboración en el desarrollo de software.

U

incertidumbre

Un concepto que hace referencia a información imprecisa, incompleta o desconocida que puede socavar la fiabilidad de los modelos predictivos de ML. Hay dos tipos de incertidumbre: la incertidumbre epistémica se debe a datos limitados e incompletos, mientras que la incertidumbre aleatoria se debe al ruido y la aleatoriedad inherentes a los datos. Para más información, consulte la guía Cuantificación de la incertidumbre en los sistemas de aprendizaje profundo.

tareas indiferenciadas

También conocido como tareas arduas, es el trabajo que es necesario para crear y operar una aplicación, pero que no proporciona un valor directo al usuario final ni proporciona una ventaja competitiva. Algunos ejemplos de tareas indiferenciadas son la adquisición, el mantenimiento y la planificación de la capacidad.

entornos superiores

Consulte entorno.

V

succión

Una operación de mantenimiento de bases de datos que implica limpiar después de las actualizaciones incrementales para recuperar espacio de almacenamiento y mejorar el rendimiento.

control de versión

Procesos y herramientas que realizan un seguimiento de los cambios, como los cambios en el código fuente de un repositorio.

Emparejamiento de VPC

Una conexión entre dos VPCs que le permite enrutar el tráfico mediante direcciones IP privadas. Para obtener más información, consulte ¿Qué es una interconexión de VPC? en la documentación de Amazon VPC.

vulnerabilidad

Defecto de software o hardware que pone en peligro la seguridad del sistema.

W

caché caliente

Un búfer caché que contiene datos actuales y relevantes a los que se accede con frecuencia. La instancia de base de datos puede leer desde la caché del búfer, lo que es más rápido que leer desde la memoria principal o el disco.

datos templados

Datos a los que el acceso es infrecuente. Al consultar este tipo de datos, normalmente se aceptan consultas moderadamente lentas.

función de ventana

Función SQL que hace un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para las tareas de procesamiento, como calcular una media móvil o acceder al valor de las filas en función de la posición relativa de la fila actual.

carga de trabajo

Conjunto de recursos y código que ofrece valor comercial, como una aplicación orientada al cliente o un proceso de backend.

flujo de trabajo

Grupos funcionales de un proyecto de migración que son responsables de un conjunto específico de tareas. Cada flujo de trabajo es independiente, pero respalda a los demás flujos de trabajo del proyecto. Por ejemplo, el flujo de trabajo de la cartera es responsable de priorizar las aplicaciones, planificar las oleadas y recopilar los metadatos de migración. El flujo de trabajo de la cartera entrega estos recursos al flujo de trabajo de migración, que luego migra los servidores y las aplicaciones.

WORM

Consulte escritura única y lectura múltiple.

WQF

Consulte AWS Workload Qualification Framework.

escritura única y lectura múltiple (WORM)

Modelo de almacenamiento que escribe los datos una sola vez y evita que se eliminen o modifiquen. Los usuarios autorizados pueden leer los datos tantas veces como sea necesario, pero no los pueden cambiar. Esta infraestructura de almacenamiento de datos se considera inmutable.

Z

ataque de día cero

Ataque, normalmente de malware, que se aprovecha de una vulnerabilidad de día cero.

vulnerabilidad de día cero

Un defecto o una vulnerabilidad sin mitigación en un sistema de producción. Los agentes de amenazas pueden usar este tipo de vulnerabilidad para atacar el sistema. Los desarrolladores suelen darse cuenta de la vulnerabilidad a raíz del ataque.

peticiones desde cero

Proporcionar a un LLM instrucciones para llevar a cabo una tarea, pero sin ejemplos (pasos) que puedan ayudar a guiarlo. El LLM debe usar los conocimientos del entrenamiento previo para

llevar a cabo la tarea. La eficacia de la petición desde cero depende de la complejidad de la tarea y de la calidad de la petición. Consulte también peticiones con pocos pasos.

aplicación zombi

Aplicación que utiliza un promedio de CPU y memoria menor al 5 por ciento. En un proyecto de migración, es habitual retirar estas aplicaciones.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.