



Rediseñe la plataforma de sus aplicaciones de mainframe mediante una base de datos compartida de IBM Db2 for z/OS para una migración gradual

AWS Guía prescriptiva



AWS Guía prescriptiva: Rediseñe la plataforma de sus aplicaciones de mainframe mediante una base de datos compartida de IBM Db2 for z/OS para una migración gradual

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Introducción	1
Resultados empresariales	2
AWS Mainframe Modernization	4
Recreación de plataformas	4
Refactorización automatizada	5
Ventajas de cambiar de plataforma	5
El proceso de transformación	7
Planificación	8
Descubrimiento de aplicaciones	8
Dependencias de datos	8
Punto de referencia de capacidad	9
Planificación de oleaje	12
Construcción	12
Coherencia de las aplicaciones	13
Arquitectura	14
Ejecutar	16
Compromiso en dos fases (2 piezas)	17
Infraestructura de ejecución	18
Testeo	19
Entorno de origen	20
Entorno de destino	20
Análisis	20
Probando su aplicación en AWS Mainframe Modernization	21
Transición	23
Provisión	23
Transmite en directo	24
Reversión	24
Concluir	24
Arquitectura	24
Prácticas recomendadas	26
Latencia de la red	26
Seguridad	27
Gobernanza de aplicaciones	27
Elasticidad	28

Pasos siguientes	29
Recursos	31
AWS documentación	31
Referencias de Rocket Software	31
Referencias de IBM	31
Tools (Herramientas)	31
AWS Patrones y guías de orientación prescriptiva	31
Historial de documentos	33
Glosario	34
#	34
A	35
B	38
C	40
D	43
E	48
F	50
G	52
H	53
I	55
L	57
M	58
O	63
P	66
Q	69
R	69
S	72
T	76
U	78
V	78
W	79
Z	80
.....	lxxxi

Cambie la plataforma de sus aplicaciones de mainframe mediante una base de datos IBM Db2 for compartida para z/OS una migración gradual

Luis Gustavo Dantas y André Botura, Amazon Web Services (AWS)

Mayo de 2025 (historia [del documento](#))

En el panorama de la tecnología empresarial en constante cambio, la modernización de los mainframes se ha convertido en un requisito fundamental para las organizaciones que necesitan seguir siendo competitivas y ágiles. Esta transformación no consiste simplemente en reemplazar los sistemas antiguos por otros nuevos; se trata de una evolución estratégica que cierra la brecha entre las bases sólidas y confiables del pasado y las posibilidades dinámicas e innovadoras del futuro.

El mainframe, que alguna vez fue el líder indiscutible de la informática empresarial, se encuentra ahora en un punto de inflexión. Su incomparable potencia de procesamiento y sus características de seguridad lo han mantenido relevante durante décadas, pero las empresas actuales necesitan sistemas que puedan integrarse sin problemas con los servicios en la nube, admitir aplicaciones móviles y aprovechar el poder de la inteligencia artificial y el análisis de macrodatos.

La modernización no siempre requiere una migración completa desde los mainframes. Algunas organizaciones están optando por enfoques híbridos que aprovechan las ventajas de los entornos de mainframe y de nube. Esta estrategia les permite mantener las aplicaciones heredadas fundamentales mientras realizan la transición gradual a plataformas más modernas. Esta transición tecnológica implica algo más que actualizaciones del sistema; requiere transformar la cultura y las habilidades organizacionales. A medida que las empresas se modernizan, invierten tanto en nuevas tecnologías como en su fuerza laboral, reduciendo las brechas generacionales y promoviendo el aprendizaje y la innovación continuos.

Esta guía analiza una estrategia de migración gradual que equilibra los beneficios de los sistemas mainframe con las ventajas de las tecnologías de nube modernas. Este enfoque de cambio de plataforma gradual migra primero la capa de aplicaciones y, al mismo tiempo, mantiene la conectividad con su z/OS base de datos IBM Db2 for existente para agilizar el proceso de transición y minimizar las interrupciones en las operaciones empresariales críticas a la vez que adopta nuevas capacidades en la nube.

Esta guía está diseñada para los responsables de la toma de decisiones técnicas y los equipos de implementación que participan en las iniciativas de modernización del mainframe. Las audiencias principales incluyen arquitectos empresariales y de soluciones, directores de proyectos técnicos y líderes de programas de modernización que necesitan comprender los aspectos estratégicos y técnicos de la replataforma de mainframe. El contenido es igualmente valioso para los equipos de implementación, incluidos los desarrolladores de aplicaciones de mainframe AWS o los ingenieros de nube, administradores de bases de datos e DevOps ingenieros responsables de ejecutar la implementación de la modernización.

Resultados empresariales

Las empresas tienen muchos motivos de peso para actualizar sus aplicaciones antiguas. Este proceso crea una sensación de urgencia en todos los sectores. Cuando los expertos de más edad se jubilan, dejan una importante brecha de conocimiento, lo que hace que sea crucial modernizar los sistemas antes de que se pierda esta experiencia. Además, las empresas se sienten impulsadas por la necesidad de reducir los costos, aumentar la agilidad y responder rápidamente a las condiciones del mercado que cambian rápidamente.

El impulso hacia la transformación digital se ve intensificado aún más por las tecnologías emergentes y la demanda de experiencias de cliente mejoradas. Estos factores, combinados con los riesgos asociados al mantenimiento de sistemas complejos, están impulsando a las organizaciones a actuar con rapidez para modernizar su infraestructura de TI.

La modernización de los mainframes, en particular, presenta un delicado equilibrio. Las empresas deben preservar la estabilidad y la seguridad que caracterizan a los mainframes y, al mismo tiempo, aprovechar la flexibilidad y la escalabilidad que ofrecen las arquitecturas modernas. Este proceso implica decisiones complejas sobre qué aplicaciones migrar, cuáles reescribir y cuáles conservar en el mainframe.

Los principales impulsores de la modernización incluyen la agilidad y la reducción de costos:

- **Agilidad y tiempo de comercialización.** Los sistemas modernos permiten procesos de adquisición más rápidos y respuestas más rápidas a las cambiantes demandas del mercado. La adopción de estas DevOps SysOps prácticas puede mejorar considerablemente la productividad y las velocidades de despliegue.
- **Reducción de costos.** La modernización a menudo conduce a una reducción de los costos de infraestructura a través de:

- Pay-as-you-go modelos, que alinean los costos con el uso real.
- Reducción de las tasas de licencia asociadas a los sistemas antiguos.
- Elasticidad mejorada, lo que proporciona una mejor asignación de recursos.
- Configuraciones activas-activas y de alta disponibilidad, que mejoran la resiliencia del sistema y, al mismo tiempo, optimizan la utilización de los recursos.

En función de estos factores empresariales, la replataforma de las aplicaciones COBOL se considera un enfoque estratégico de la modernización. Puede utilizar una base de datos compartida para seguir una ruta de migración gradual que equilibre la necesidad de modernización con el imperativo de mantener la continuidad empresarial. Este método le permite aprovechar los beneficios de las arquitecturas modernas y, al mismo tiempo, preservar la confiabilidad de sus aplicaciones COBOL. Como resultado, puede lograr agilidad, rentabilidad e innovación y, al mismo tiempo, mitigar los riesgos asociados a las transiciones abruptas y a gran escala. El enfoque de base de datos DB2 compartida que se describe en esta guía proporciona un puente entre los sistemas heredados y las plataformas modernas, y permite un proceso de modernización más fluido y controlado.

En esta guía:

- [AWS Mainframe Modernization](#)
- [El proceso de transformación](#)
- [Prácticas recomendadas](#)
- [Pasos siguientes](#)
- [Recursos](#)
- [Historial de revisión](#)

AWS Mainframe Modernization

Note

AWS Mainframe Modernization El servicio (experiencia de entorno de ejecución gestionado) ya no está abierto a nuevos clientes. Para obtener funcionalidades similares a las AWS Mainframe Modernization de Service (experiencia en entornos de ejecución gestionados), explore AWS Mainframe Modernization Service (experiencia autogestionada). Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad de AWS Mainframe Modernization](#).

El [AWS Mainframe Modernization servicio](#) le permite migrar sus aplicaciones de mainframe heredadas a un entorno nativo de la nube, preservar la lógica empresarial y las inversiones existentes, utilizar herramientas automatizadas y servicios de tiempo de ejecución gestionados, optimizar el rendimiento de las aplicaciones y reducir los costes operativos. Este servicio agiliza el proceso de modernización para que pueda aprovechar el poder de la nube y, al mismo tiempo, conservar el valor de sus principales sistemas de mainframe. AWS ofrece dos enfoques clave para la modernización de los mainframes: la replataforma y la refactorización automatizada.

Recreación de plataformas

[AWS Mainframe Modernization La replataforma con Rocket Software](#) (anteriormente Micro Focus) ofrece una potente opción de cambio de plataforma para las empresas que desean migrar sus aplicaciones de mainframe a la nube con una interrupción mínima. Esta solución le permite volver a compilar y ejecutar sus PL/I aplicaciones y COBOL existentes sin necesidad de realizar cambios importantes en el AWS código.

Entre las principales ventajas de la solución AWS Replatform with Rocket Software se incluyen las siguientes:

- Preservación de la lógica empresarial y las inversiones existentes
- Reducción del riesgo y reducción del tiempo de comercialización
- Escalabilidad y rendimiento mejorados en la infraestructura AWS
- Acceso a prácticas y herramientas de desarrollo modernas

Puede utilizar esta solución para mantener sus lenguajes de programación de mainframe conocidos y, al mismo tiempo, aprovechar la flexibilidad, la rentabilidad y la innovación del. Nube de AWS

Refactorización automatizada

Si busca un enfoque más transformador que el cambio de plataforma, puede utilizar [AWS Blu Age](#), que ofrece una refactorización automática de las aplicaciones de mainframe para convertirlas en aplicaciones nativas de la nube basadas en Java. Esta solución le ayuda a modernizar sus sistemas heredados de manera más integral y a convertirlos en aplicaciones que pueden aprovechar al máximo las tecnologías nativas de la nube.

Las principales ventajas de AWS Blu Age incluyen:

- Conversión de código heredado en aplicaciones Java modernas y fáciles de mantener
- Transformación automatizada que reduce el esfuerzo manual y los posibles errores
- Creación de aplicaciones nativas de la nube optimizadas para Servicios de AWS
- Agilidad mejorada y una integración más sencilla con las tecnologías modernas

AWS Blu Age le ayuda a migrar sus aplicaciones y a prepararlas para la nube, a fin de abrir nuevas posibilidades de innovación y crecimiento. Para obtener más información sobre este enfoque, consulte [Refactorizar aplicaciones automáticamente con AWS Blu Age](#) en la AWS Mainframe Modernization documentación.

Ventajas de cambiar de plataforma

Esta guía analiza un enfoque para cambiar la plataforma de las aplicaciones COBOL de un mainframe. Este enfoque tiene como objetivo modernizar los sistemas heredados y, al mismo tiempo, conservar temporalmente IBM Db2 para z/OS agilizar el proceso de transición. Al mantener inicialmente la estructura de base de datos existente, puede reducir la complejidad y el riesgo durante la migración. Este enfoque gradual le ayuda a beneficiarse de la escalabilidad y la rentabilidad del sistema, a la Nube de AWS vez que preserva la integridad de los datos críticos. Entre las ventajas del cambio de plataforma por fases se incluyen las siguientes:

- Modernización acelerada: la replataforma y la refactorización suelen requerir menos tiempo y recursos en comparación con la reinención de una aplicación antigua en la nube, ya que no implican reescribir toda la aplicación. Este enfoque también permite una transición más gradual

que permite a las organizaciones modernizarse a su propio ritmo y, al mismo tiempo, beneficiarse inmediatamente de la escalabilidad y la rentabilidad de la Nube de AWS

- **Mitigación de riesgos:** la replataforma ofrece varias ventajas en comparación con la refactorización para muchas organizaciones. Las empresas pueden mantener sus PL/I bases de código y COBOL actuales, preservar años de lógica empresarial y minimizar el riesgo asociado a los cambios de código a gran escala.
- **Continuidad de los datos y migración gradual:** una ventaja importante de la replataforma es la opción de mantener inicialmente los datos en Db2 y mantenerlos en su formato de datos original. z/OS Esta estrategia evita la necesidad de procesos de migración de datos inmediatos, complejos y potencialmente riesgosos. Al mantener los datos en su entorno original durante la fase inicial, puede preservar la integridad de los datos, reducir el tiempo de inactividad y minimizar el riesgo de pérdida o corrupción de los datos durante el proceso de modernización. Como segundo paso, puede planificar una migración de datos controlada y gradual a bases de datos nativas de la nube, que implique pruebas y validaciones exhaustivas mientras la aplicación sigue ejecutándose en el entorno con la nueva plataforma.
- **Flexibilidad y preparación para el futuro:** para las empresas que tienen inversiones importantes en habilidades y aplicaciones de mainframe, la replataforma proporciona un camino pragmático hacia la modernización que equilibra la innovación con la continuidad. Ofrece la flexibilidad necesaria para conservar inicialmente las estructuras de datos y los métodos de acceso críticos y, al mismo tiempo, sienta las bases para los futuros esfuerzos de modernización, incluida la eventual migración de datos a soluciones totalmente nativas de la nube.

Las organizaciones pueden seguir el enfoque de replataforma para modernizarse a su propio ritmo y abordar las necesidades inmediatas, al tiempo que planifican los objetivos de transformación digital a largo plazo. Este enfoque también brinda a las empresas la oportunidad de capacitar a su personal sobre los servicios nativos de la nube.

El proceso de transformación

La modernización del mainframe es un paso fundamental para las organizaciones que desean aprovechar los beneficios de la computación en nube y, al mismo tiempo, conservar sus valiosas aplicaciones heredadas. Esta transformación presenta desafíos importantes. Las aplicaciones de mainframe suelen estar muy acopladas y tienen intrincadas interdependencias que han evolucionado a lo largo de décadas de funcionamiento. Esta complejidad exige un enfoque cuidadoso y metódico de la modernización.

Las organizaciones deben superar las siguientes fases clave para lograr una transición exitosa:

- **Planificación:** esta fase implica un descubrimiento exhaustivo de los sistemas existentes y la priorización de las iniciativas de modernización. Las organizaciones evalúan su infraestructura actual, identifican las aplicaciones críticas y determinan qué sistemas deben modernizarse primero.
- **Creación:** durante esta etapa, las organizaciones crean procesos para migrar las aplicaciones y desarrollar nuevos sistemas e infraestructuras. Esto implica diseñar e implementar la arquitectura modernizada y compilar el código fuente.
- **Ejecución:** este paso consiste en crear los entornos de ejecución para alojar las aplicaciones reconfiguradas. Implica configurar el hardware, el software y la infraestructura de nube necesarios para respaldar los sistemas modernizados y garantizar que puedan funcionar de manera eficiente en el nuevo entorno.
- **Pruebas:** esta fase incluye una validación rigurosa de los sistemas modernizados para verificar que se hayan cumplido todos los requisitos funcionales y de rendimiento. Se llevan a cabo pruebas exhaustivas para verificar la integridad de los datos, la compatibilidad del sistema y el rendimiento general del nuevo entorno.
- **Transición:** la fase final se centra en implementar estrategias para una transición fluida y controlar el cambio del mainframe heredado al entorno modernizado. Esto incluye una planificación cuidadosa del programa de migración y los planes de contingencia para minimizar las interrupciones en las operaciones comerciales.

En las siguientes secciones se analizan estas fases en detalle:

- [Planificación](#)
- [Edificio](#)

- [En ejecución](#)
- [Pruebas](#)
- [Transición](#)

Planificación

Para cumplir con eficacia los requisitos de las aplicaciones antiguas de mainframe, las organizaciones suelen comenzar con una evaluación exhaustiva de su entorno de mainframe.

Descubrimiento de aplicaciones

Una herramienta poderosa en esta fase inicial es el [Rocket Enterprise Analyzer](#), que proporciona información detallada sobre la estructura, las dependencias y la complejidad de las aplicaciones de mainframe. Esta herramienta le ayuda a determinar el alcance de su esfuerzo de modernización, los posibles riesgos y las oportunidades de optimización.

Un aspecto crucial que hay que descubrir es la intrincada red de dependencias de datos que existe en los sistemas de mainframe. Estas dependencias suelen estar ocultas bajo capas de código heredado y pueden tener un impacto significativo en los esfuerzos de modernización. Al trazar un mapa de cómo interactúan las diferentes aplicaciones y módulos con las distintas fuentes de datos, podrá comprender mejor los posibles efectos de cualquier cambio que tenga previsto implementar.

Dependencias de datos

Una evaluación exhaustiva de las dependencias de los datos puede revelar información fundamental sobre el flujo, la calidad y el gobierno de los datos en su entorno de mainframe. Este conocimiento tiene un valor incalculable a la hora de planificar estrategias de migración de datos, garantizar la integridad de los datos durante la modernización e identificar oportunidades de optimización de los datos. Al obtener una imagen clara de sus datos, puede tomar decisiones más informadas sobre qué enfoques de modernización serán más eficaces y menos perjudiciales para sus operaciones actuales.

Un análisis descendente que identifique el uso de las tablas por transacciones o por tareas relacionadas con el lenguaje de control de tareas (JCL) es fundamental para planificar y priorizar una oleada. Este enfoque aclara las relaciones entre los diferentes componentes de los sistemas de mainframe y le ayuda a desarrollar un enfoque estratégico y gradual de la modernización. Al

identificar a qué tablas se accede con más frecuencia y mediante qué procesos, puede priorizar sus esfuerzos de modernización: puede centrarse primero en las áreas de mayor impacto y garantizar una transición más fluida con una interrupción mínima de las operaciones comerciales críticas.

Además de utilizar Rocket Enterprise Analyzer para descubrir las dependencias de los datos, muchas organizaciones también utilizan sus propias soluciones personalizadas para obtener una visión más profunda de sus entornos de mainframe. Estas herramientas internas suelen aprovechar la gran cantidad de información disponible en el catálogo de IBM Db2 y en los registros del System Management Facility (SMF).

Punto de referencia de capacidad

Un paso a la hora de planificar su proyecto de cambio de plataforma de mainframe consiste en recopilar información detallada sobre el consumo actual de la carga de trabajo. Estos datos le ayudarán a predecir y aprovisionar con precisión la capacidad inicial requerida en su entorno de nube objetivo. Por ejemplo, le recomendamos que recopile datos sobre el consumo de millones de instrucciones por segundo (MIPS) por hora, tanto para las transacciones en línea como para las transacciones por lotes, de los trabajos del Sistema de Control de Información del Cliente (CICS) o del Sistema de Gestión de la Información (IMS) y del lenguaje de control de tareas (JCL) de IBM.

IBM ofrece una amplia gama de [modelos de precios](#) para el MIPS en la computación de mainframe, y muchos de estos modelos se centran en los picos de uso. Entre estos modelos basados en picos, el más común es el pico continuo de cuatro horas.

Los costos del mainframe incluyen cinco áreas clave que tienen un impacto significativo en los gastos generales:

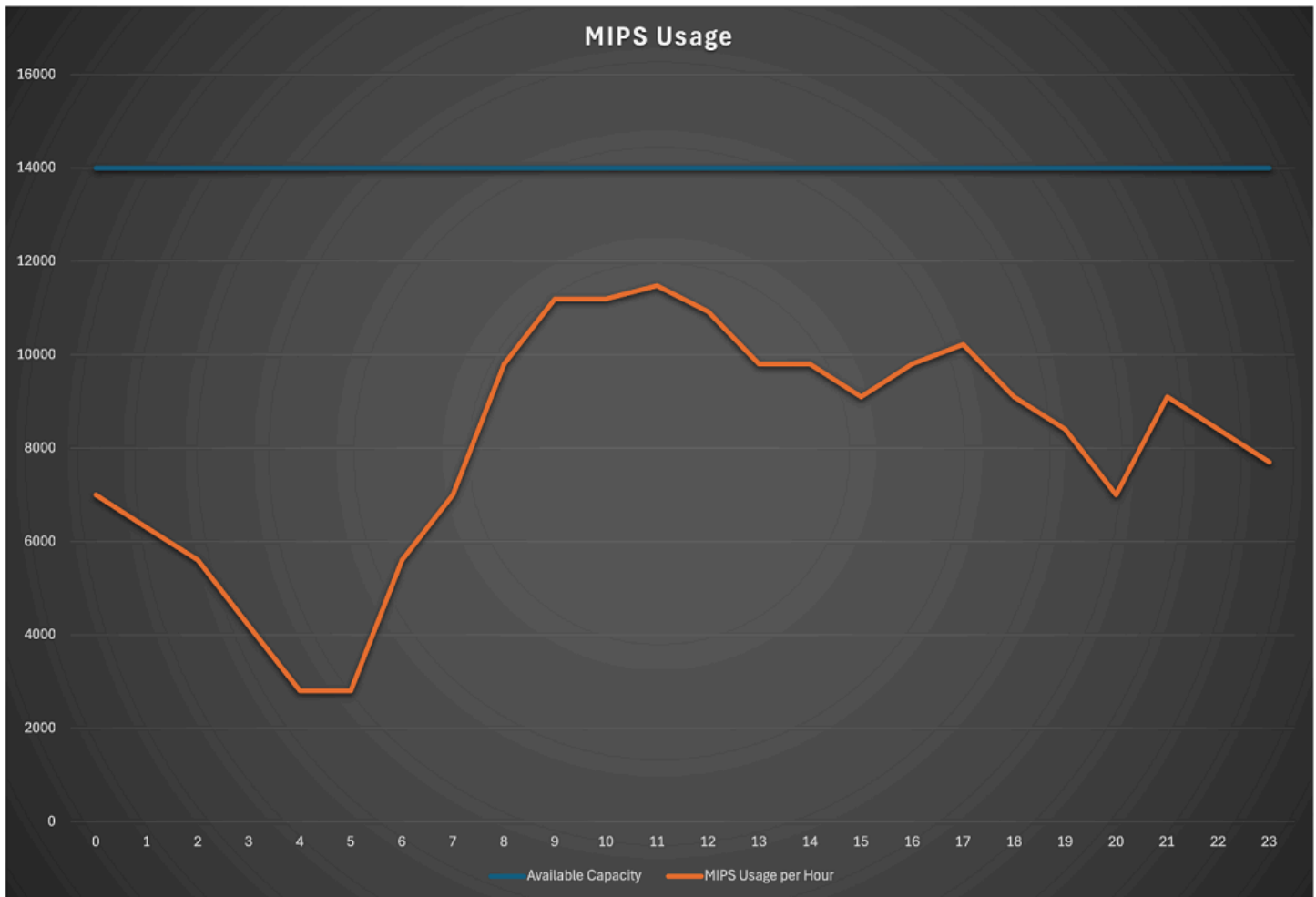
- Las licencias de software suelen ser un componente importante. Abarca los sistemas operativos, el middleware, las bases de datos y diversas aplicaciones, y los costos a veces están relacionados con la capacidad o el uso de la máquina.
- Los gastos de hardware incluyen la compra o el arrendamiento iniciales del equipo de mainframe, el mantenimiento continuo y las actualizaciones.
- Los costos de almacenamiento pueden ser considerables debido a la gran cantidad de datos administrados que se administran e incluyen sistemas de discos, bibliotecas de cintas y el software de administración asociado.
- Los gastos de personal cubren los salarios de los profesionales especializados en mainframe, como los programadores de sistemas y los administradores de bases de datos.

- Las medidas de recuperación ante desastres y continuidad empresarial, incluidos los sistemas de respaldo, el hardware redundante y las instalaciones de recuperación externas, representan una inversión importante para garantizar una alta disponibilidad y una recuperación rápida.

Estas cinco categorías de costos, combinadas con los cargos basados en el MIPS, constituyen el núcleo de la mayoría de los presupuestos de mainframe. Sin embargo, sus proporciones relativas pueden variar considerablemente según el tamaño de la organización, el sector y los patrones específicos de uso del mainframe.

Los datos de MIPS por hora son cruciales para obtener una comprensión completa de los patrones de carga de trabajo y el rendimiento de su mainframe. A diferencia de los promedios diarios o mensuales, los datos por hora proporcionan información detallada que revela las fluctuaciones matizadas en la utilización de los recursos del sistema a lo largo del día. Este nivel de detalle es inestimable para evaluar con precisión las necesidades de rendimiento y capacidad de su aplicación en la nube.

Al analizar los datos del MIPS por hora, puede identificar los períodos de uso máximo, detectar tendencias y detectar posibles cuellos de botella que podrían ocultarse en los datos agregados, como se muestra en el siguiente diagrama. Esta granularidad permite una planificación de la capacidad más precisa, ayuda a optimizar la asignación de recursos y puede suponer un ahorro de costes y una mejora de la eficiencia del sistema.



Los datos MIPS por hora también sirven como una herramienta esencial de referencia del rendimiento. Establece una línea base detallada del rendimiento del sistema, lo que resulta especialmente valioso a la hora de planificar o evaluar cambios en el sistema, como migraciones o actualizaciones. Al comparar los datos MIPS horarios previos y posteriores al cambio, puede medir con precisión el impacto de estas modificaciones en el rendimiento del sistema y garantizar que su mainframe siga satisfaciendo las necesidades de su organización.

Para recopilar datos de MIPS por hora, tiene varias opciones. Un enfoque consiste en utilizar los registros SMF directamente. Estos registros proporcionan una gran cantidad de información sobre la actividad del sistema y el uso de los recursos. Como alternativa, puede utilizar herramientas especializadas, como la herramienta IBM Sub-Capacity Reporting Tool (SCRT), que puede simplificar el proceso de recopilación y análisis de los datos del MIPS.

Independientemente del método que elija, es importante recopilar los datos durante un período prolongado, idealmente, varios meses. Este período de recopilación prolongado le permite

tener en cuenta las variaciones cíclicas de su carga de trabajo, como los picos de end-of-month procesamiento o las fluctuaciones estacionales. Al capturar estos patrones a largo plazo, podrá obtener una imagen más precisa y completa de las características de rendimiento de su mainframe, lo que le permitirá tomar decisiones mejor fundamentadas y gestionar la capacidad de forma más eficaz.

Planificación de oleaje

Puede utilizar la información recopilada para priorizar estratégicamente sus iniciativas de cambio de plataforma de mainframe. Un enfoque prudente consiste en empezar con cargas de trabajo menos críticas, como las transacciones comerciales no esenciales o los trabajos por lotes, para que los equipos puedan adquirir experiencia y perfeccionar los procesos con un riesgo mínimo para las operaciones esenciales. Además, considerar las cargas de trabajo de solo lectura como primeras candidatas para la migración puede ser ventajoso, ya que estas cargas de trabajo suelen implicar menos complejidad y un menor riesgo de inconsistencias en los datos. Este enfoque le permite generar confianza e impulsar sus esfuerzos de cambio de plataforma.

Además, agrupar las cargas de trabajo que comparten tablas de Db2 para operaciones de escritura o actualización puede agilizar el proceso de migración. Al identificar estas cargas de trabajo interconectadas, puede planificar oleadas de migración cohesivas que mantengan la integridad de los datos y minimicen la necesidad de soluciones provisionales complejas. Esta estrategia no solo reduce el riesgo de conflictos de datos, sino que también optimiza el cronograma general de cambio de plataforma al abordar los componentes relacionados simultáneamente. En última instancia, este enfoque de priorización basado en los datos garantiza una consideración equilibrada de la criticidad, la complejidad y la interdependencia, y conduce a un proceso de modernización del mainframe más eficiente y exitoso.

Construcción

El uso de una base de datos Db2 compartida permite la ejecución simultánea de aplicaciones idénticas o consistentes tanto en entornos de mainframe como de nube. Este enfoque ofrece varias ventajas al mantener la misma versión de la aplicación en ambas plataformas y proporciona una mayor flexibilidad y confiabilidad en sus operaciones.

Una ventaja clave de esta estrategia es la capacidad de implementar un plan de reversión eficaz. Si surgen problemas durante la migración o la implementación, tener la misma versión de la aplicación

permite volver al estado anterior sin problemas y minimiza el tiempo de inactividad y las posibles inconsistencias en los datos.

Coherencia de las aplicaciones

La duplicación de los componentes de la aplicación desde un administrador de control de código fuente distribuido al mainframe es un enfoque estratégico durante el proceso de cambio de plataforma. Este método admite el uso de herramientas modernas de administración del código fuente y, al mismo tiempo, mantiene la sincronización con el entorno del mainframe. Este proceso de duplicación es temporal y solo dura hasta que la carga de trabajo esté completamente operativa en la producción en la plataforma distribuida.

Al migrar el código fuente de la aplicación rediseñada a una herramienta de gestión de cambios distribuida, puede aprovechar las diversas ventajas que ofrecen los gestores de código fuente modernos. Entre ellos se incluyen:

- **Colaboración mejorada:** las herramientas distribuidas suelen ofrecer un mejor soporte para la colaboración en equipo, ya que incluyen funciones como las solicitudes de cambios, las revisiones del código y las estrategias de ramificación.
- **Control de versiones mejorado:** los sistemas modernos ofrecen un control de versiones más detallado y facilitan el seguimiento de los cambios y la administración de las diferentes versiones del código.
- **Integración con los CI/CD canales:** muchas herramientas distribuidas se integran a la perfección con los canales de integración y despliegue continuos (CI/CD), lo que agiliza el proceso de desarrollo.
- **Mejor visibilidad y trazabilidad:** estas herramientas suelen proporcionar cuadros de mando y capacidades de generación de informes superiores, además de ofrecer una mayor visión del proceso de desarrollo.
- **Support for modern development practices:** Los sistemas distribuidos suelen ser más adecuados para metodologías y DevOps prácticas ágiles.

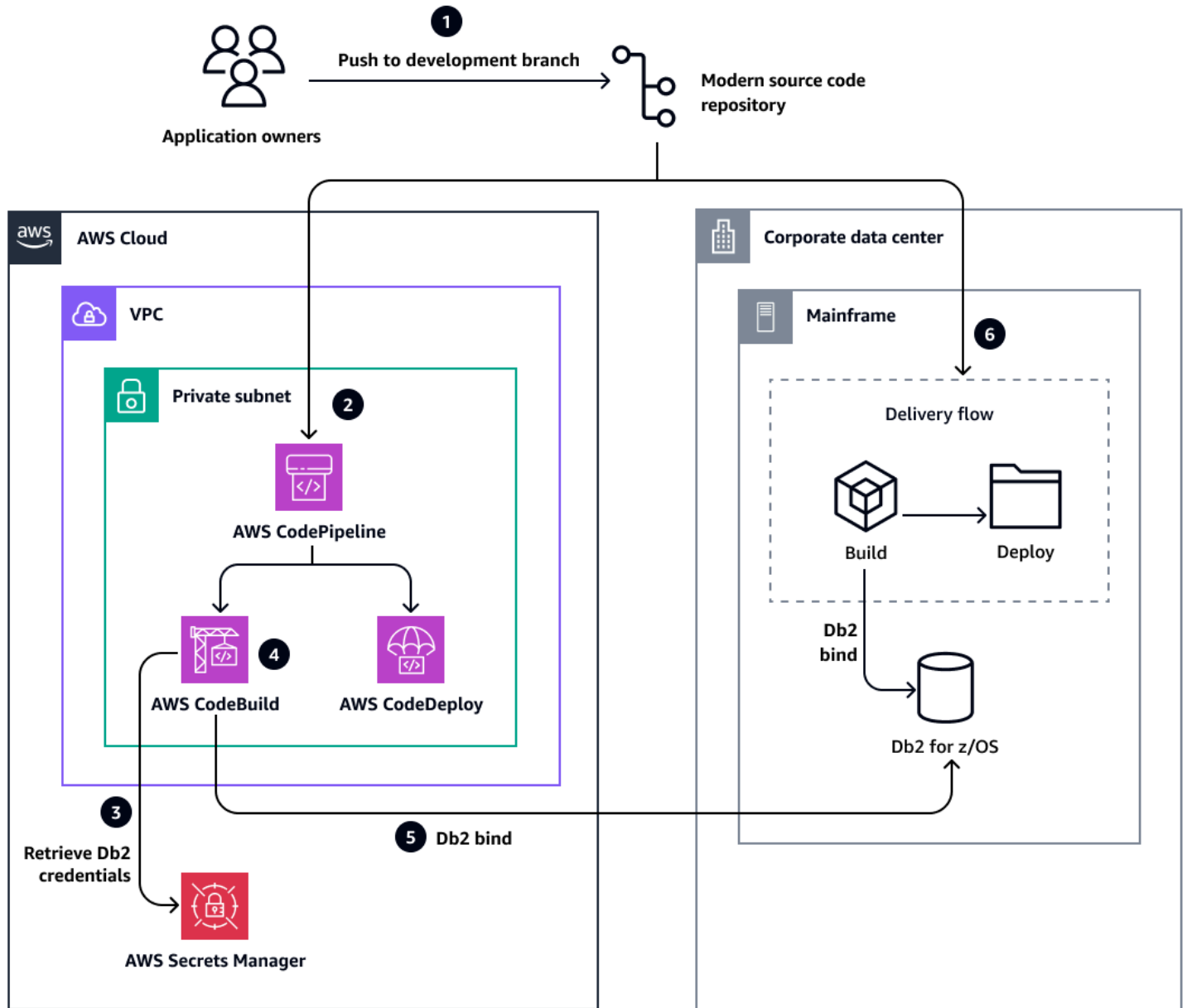
El proceso de duplicación implica la sincronización del código desde el administrador de control de código fuente distribuido hasta el mainframe. Esto garantiza que ambos entornos permanezcan consistentes durante el período de transición. Sin embargo, debe implementar la duplicación como una sincronización unidireccional, en la que las actualizaciones fluyan del sistema distribuido al mainframe, en lugar de hacerlo de forma bidireccional. Este enfoque mantiene la coherencia y evita

los posibles conflictos que podrían surgir a causa de las actualizaciones simultáneas en ambos entornos.

Al adoptar esta estrategia de duplicación, puede trasladar gradualmente sus esfuerzos de desarrollo a la plataforma distribuida y, al mismo tiempo, garantizar que se mantenga el entorno de mainframe. up-to-date Esto proporciona una transición más fluida y una red de seguridad durante el proceso de cambio de plataforma. Cuando la carga de trabajo sea completamente funcional y estable en el entorno de producción distribuida, podrá eliminar gradualmente el proceso de duplicación y completar la migración al moderno sistema de administración de código fuente.

Arquitectura

El siguiente diagrama muestra cómo un sistema de administración de código fuente distribuido puede duplicar los componentes de la aplicación y mantener la sincronización entre los entornos de mainframe Nube de AWS y los de mainframe. El Nube de AWS entorno utiliza CI/CD servicios como [AWS CodeBuild](#), [AWS CodePipeline](#), y [AWS CodeDeploy](#) para crear e implementar la aplicación.



En este flujo de trabajo:

1. Los propietarios de la aplicación envían una nueva versión de la aplicación a la rama de desarrollo del repositorio de código fuente.
2. Se activa la nueva versión AWS CodePipeline.
3. AWS CodeBuild recupera las credenciales de Db2 de [AWS Secrets Manager](#)
4. CodeBuild compila la aplicación.
5. CodeBuild usa Db2 para z/OS vincular la aplicación.

6. El flujo de entrega del mainframe también crea e implementa la aplicación.

Ejecutar

Para garantizar un rendimiento óptimo y una baja latencia entre la aplicación basada en la nube y la base de datos local, le recomendamos que la implemente. [AWS Direct Connect](#) Este servicio proporciona una conexión de red dedicada entre AWS el centro de datos de su organización y ofrece un rendimiento de red más uniforme y una latencia reducida en comparación con las conexiones basadas en Internet. Esto es especialmente importante para las operaciones de bases de datos que requieren tiempos de respuesta rápidos.

Para lograr una alta disponibilidad (HA) y elasticidad para la aplicación en la que se ejecuta AWS, puede implementar una arquitectura sólida mediante el uso de los siguientes componentes:

- Elastic Load Balancing (ELB): puede implementar un balanceador de carga para distribuir el tráfico entrante entre varias instancias de Amazon Elastic Compute Cloud (Amazon EC2) en las que se ejecute la aplicación. Esto garantiza una distribución uniforme de la carga de trabajo y proporciona un punto de entrada único para las solicitudes de los clientes.
- Grupo Auto Scaling: las instancias EC2 que alojan la aplicación se pueden organizar en un grupo de Auto Scaling. Esto permite que la infraestructura ajuste automáticamente el número de instancias en función de métricas predefinidas, como el uso de la CPU o el tráfico de red. Durante las horas punta, se pueden lanzar instancias adicionales para gestionar el aumento de la carga, mientras que durante los períodos más silenciosos, se pueden eliminar las instancias innecesarias para optimizar los costes.
- Instancias EC2: la aplicación se puede implementar en instancias EC2 del grupo Auto Scaling. Estas instancias deben distribuirse en varias zonas de disponibilidad para mejorar la tolerancia a los errores y garantizar una alta disponibilidad.
- Implementación en zonas de disponibilidad múltiples: al distribuir las instancias de la aplicación en varias zonas de disponibilidad, el sistema puede resistir el fallo de una sola zona de disponibilidad sin que ello afecte de forma significativa a la disponibilidad general.

Esta arquitectura permite que la aplicación se escale sin problemas en función de la demanda y, al mismo tiempo, mantiene una alta disponibilidad. El balanceador de carga garantiza que el tráfico se distribuya uniformemente entre las instancias en buen estado, y el grupo Auto Scaling administra la cantidad de instancias en función de la carga de trabajo real.

Para mejorar aún más la confiabilidad, puedes implementar un sistema sólido de monitoreo y alertas utilizando [Amazon CloudWatch](#) para ayudar a detectar y responder rápidamente a cualquier problema o falla de rendimiento. Además, las pruebas periódicas de las capacidades de escalado automático y los escenarios de conmutación por error garantizarán que el sistema se comporte según lo esperado en diversas condiciones de carga y posibles fallos.

Al adoptar este enfoque, puede beneficiarse de la escalabilidad y la flexibilidad de mantener, al Nube de AWS mismo tiempo, una conexión segura a su base de datos Db2 local. Esta configuración híbrida constituye un excelente camino hacia una migración completa a la nube y proporciona una transición gradual y una mitigación de los riesgos a lo largo del proceso.

Compromiso en dos fases (2 piezas)

[AWS Mainframe Modernization Replatform with Rocket Software](#) ofrece soporte para transacciones de confirmación en dos fases (2PC) mediante la implementación de una arquitectura extendida (XA). Esta capacidad es crucial para mantener la integridad de los datos en los sistemas distribuidos, especialmente en los entornos de mainframe, donde las transacciones complejas suelen abarcar varios recursos.

La arquitectura XA, que está integrada en AWS Replatform with Rocket Software, permite coordinar las transacciones entre diversos recursos, como bases de datos y colas de mensajes. Esta integración garantiza que todas las partes de una transacción distribuida se confirmen o se anulen al unísono, a fin de mantener la coherencia en todo el sistema.

El proceso 2PC consta de dos fases:

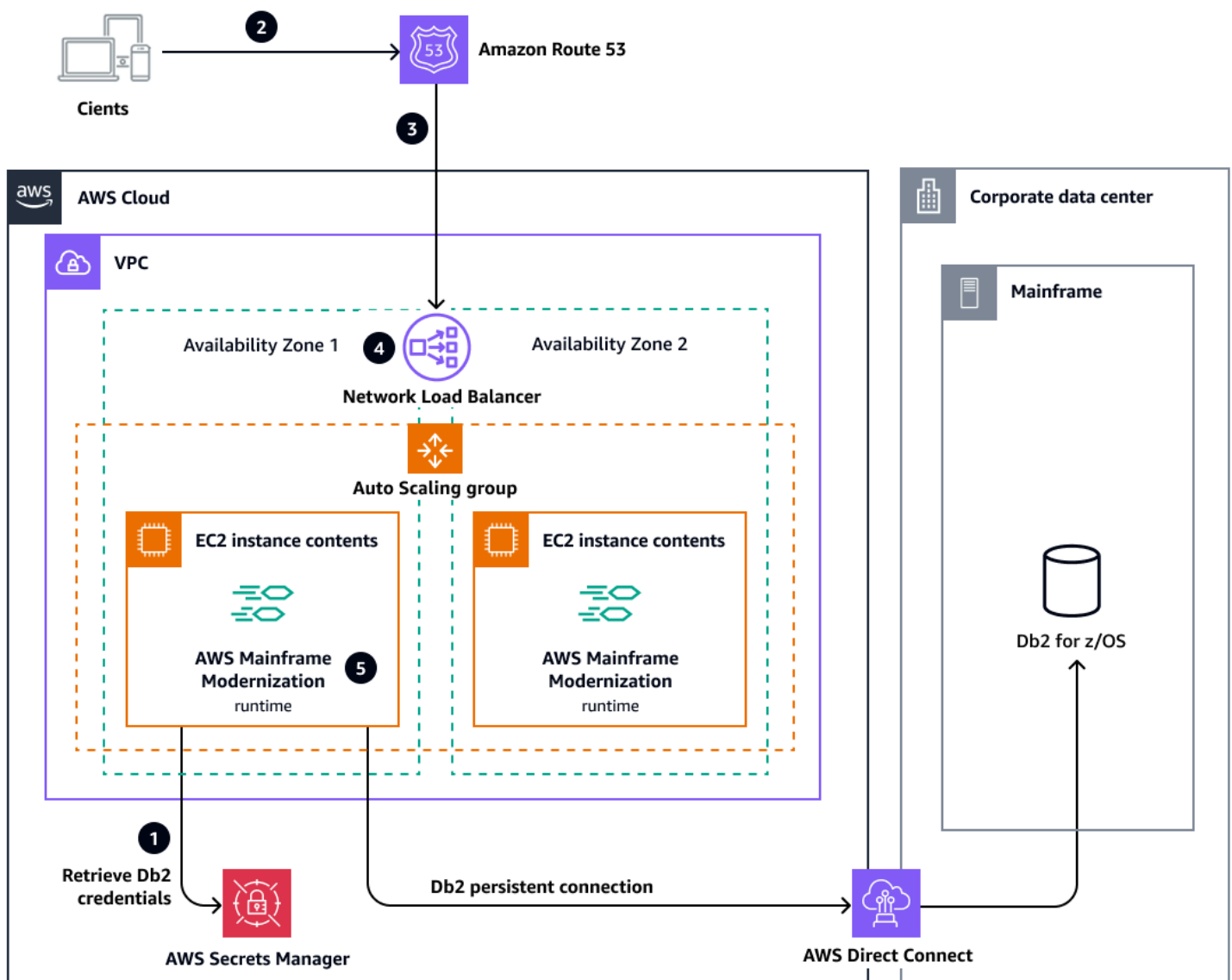
- Fase de preparación: el administrador de transacciones consulta a todos los administradores de recursos que participan en la transacción para asegurarse de que están preparados para comprometerse.
- Fase de confirmación: si todos los administradores de recursos responden positivamente, el administrador de transacciones les indica que confirmen los cambios. Si alguno de los administradores de recursos no puede comprometerse, se les indica a todos los administradores que anulen los cambios.

Al utilizar XA, AWS Replatform with Rocket Software proporciona una solución fiable y escalable para gestionar transacciones distribuidas y complejas en entornos de mainframe modernizados. Esta

función es esencial para las organizaciones que desean migrar sus aplicaciones de mainframe a la nube sin comprometer la integridad o el rendimiento de las transacciones.

Infraestructura de ejecución

El siguiente diagrama muestra un entorno elástico y de alta disponibilidad Nube de AWS que incluye dos zonas de disponibilidad, instancias EC2 en un grupo de Auto Scaling, un Network Load Balancer y una conexión dedicada entre los entornos de mainframe y AWS los de mainframe a través de él. AWS Direct Connect



En esta arquitectura:

1. Cuando se inicia el AWS Mainframe Modernization tiempo de ejecución, recupera las credenciales de Db2 for z/OS [AWS Secrets Manager](#) y abre una conexión persistente con Db2 for z/OS.

Note

AWS Mainframe Modernization El servicio (experiencia de entorno de ejecución gestionado) ya no está abierto a nuevos clientes. Para obtener funcionalidades similares a las AWS Mainframe Modernization de Service (experiencia en entornos de ejecución gestionados), explore AWS Mainframe Modernization Service (experiencia autogestionada). Los clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad de AWS Mainframe Modernization](#).

2. Los clientes enlazan la dirección de Network Load Balancer en [Amazon Route 53](#).
3. Route 53 redirige las transacciones al Network Load Balancer.
4. El Network Load Balancer distribuye las transacciones entre varias instancias de EC2.
5. La carga de trabajo en la que se ejecuta AWS Mainframe Modernization interactúa con Db2 z/OS mediante una conexión persistente. AWS Direct Connect

Testeo

Cuando se cambia la plataforma de una aplicación COBOL y se mantiene Db2 z/OS como base de datos compartida, es fundamental garantizar que el nuevo sistema funcione de forma equivalente al original. Este entorno híbrido presenta desafíos y oportunidades únicos para las pruebas. La siguiente estrategia describe un enfoque integral para las pruebas de equivalencia funcional y está diseñada para validar el rendimiento de la aplicación rediseñada, la integridad de los datos y la perfecta integración con la base de datos existente de Db2 for z/OS.

Comience por identificar los procesos empresariales y las transacciones fundamentales que deben compararse entre los sistemas. Luego, cree un plan de pruebas detallado con escenarios específicos que evalúen de manera efectiva la equivalencia funcional de estas transacciones. Por último, desarrolle conjuntos de datos de prueba completos que cubran todos los escenarios identificados y asegúrese de que sean idénticos para ambos sistemas a fin de permitir una comparación precisa.

Entorno de origen

- Instantánea inicial (primera instantánea):
 - Asegúrese de que otras aplicaciones no utilicen la tabla de datos durante la prueba, ya que esto puede afectar a la prueba de equivalencia.
 - Antes de ejecutar cualquier prueba, tome una instantánea del Db2 para ver z/OS las tablas que utiliza la transacción.
- Pruebas del sistema de origen:
 - Ejecute el conjunto completo de pruebas en la aplicación COBOL original.
 - Registre todas las transacciones, entradas y salidas.
 - Supervise el rendimiento del sistema y la utilización de los recursos.
- Instantánea de las pruebas posteriores al suministro (segunda instantánea):
 - Realice otra instantánea de la z/OS base de datos de Db2 for después de completar las pruebas del sistema de origen.

Entorno de destino

- Restablecer la base de datos:
 - Restaure la base de datos a su estado inicial mediante la primera instantánea.
- Pruebas del sistema objetivo (entorno rediseñado):
 - Ejecute el mismo conjunto de pruebas en la aplicación con la nueva plataforma.
 - Asegúrese de que todas las pruebas del sistema de destino utilicen las mismas entradas que las pruebas del sistema de origen.
 - Supervise el rendimiento del sistema y la utilización de los recursos.
- Instantánea de las pruebas posteriores a Target (tercera instantánea):
 - Realice una instantánea final de la z/OS base de datos de Db2 for después de completar las pruebas del sistema de destino.

Análisis

- Comparación y análisis:
 - Compare la segunda y la tercera instantánea para identificar cualquier discrepancia en los datos.

- Analice los resultados de las pruebas y compare los resultados de los sistemas de origen y de destino.
- Evalúe las métricas de rendimiento entre los dos entornos.
- Pruebas de integración:
 - Realice pruebas que involucren tanto a la aplicación rediseñada como a cualquier componente restante de COBOL.
 - Compruebe la perfecta interacción entre los dos entornos.
- Pruebas de recuperación y conmutación por error:
 - Escenarios de prueba en los que un entorno falla y el otro asume el control.
 - Garantice la coherencia e integridad de los datos durante las situaciones de conmutación por error.
- Pruebas de carga y stress:
 - Realice pruebas con cargas variables para evaluar el rendimiento del sistema híbrido en situaciones de stress.
 - Identifique cualquier obstáculo o problema de rendimiento en cualquiera de los entornos.
- Documentación e informes:
 - Documente todos los resultados de las pruebas, las discrepancias y las métricas de rendimiento.
 - Prepare un informe completo que compare los sistemas de origen y destino.

Probando su aplicación en AWS Mainframe Modernization

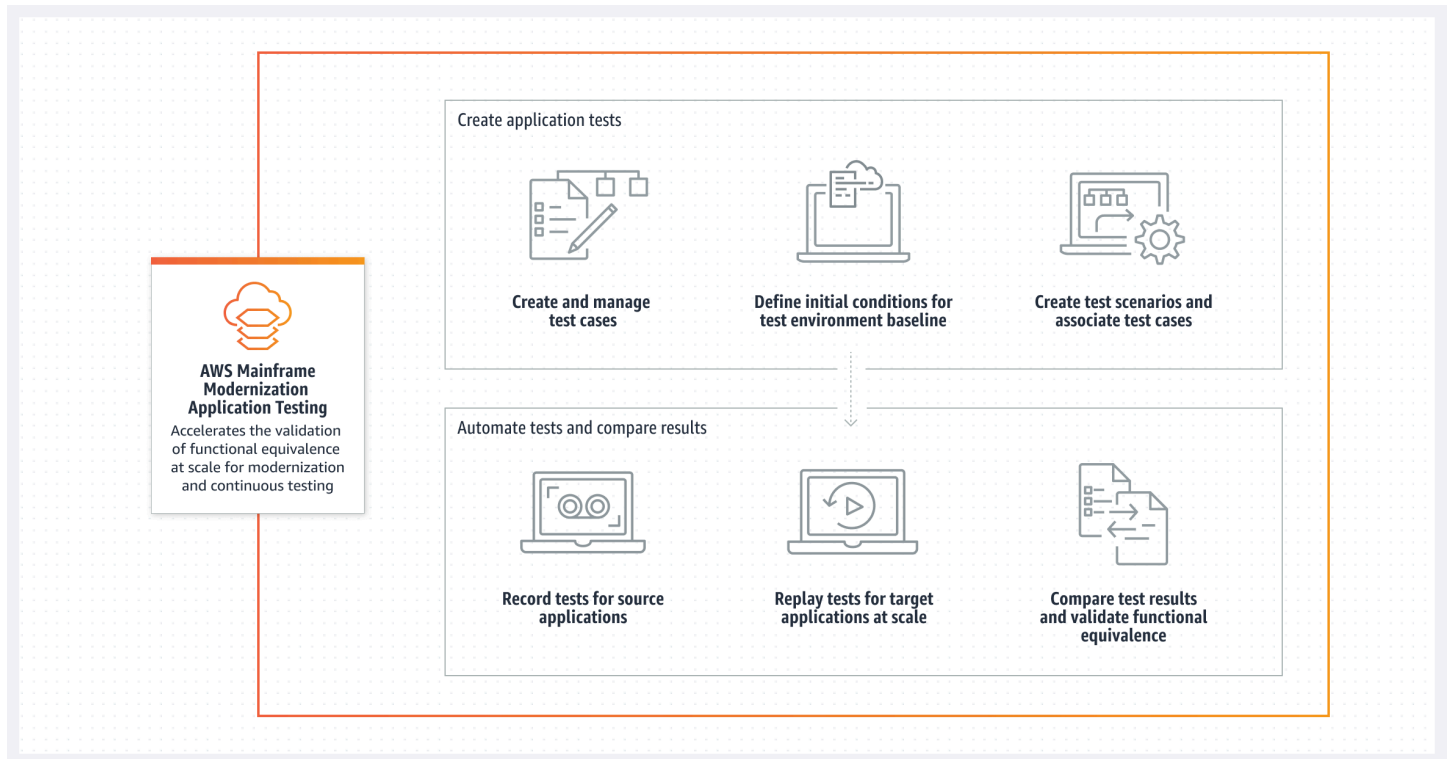
El [Prueba de aplicación de AWS Mainframe Modernization](#) servicio automatiza la ejecución de las pruebas de aplicaciones a escala. Prueba de aplicación de AWS ayuda a optimizar y reducir los costos de los proyectos de prueba y modernización de las aplicaciones de mainframe.

Note

AWS Mainframe Modernization El servicio (experiencia de entorno de ejecución gestionado) ya no está abierto a nuevos clientes. Para obtener funcionalidades similares a las AWS Mainframe Modernization de Service (experiencia en entornos de ejecución gestionados), explore AWS Mainframe Modernization Service (experiencia autogestionada). Los

clientes existentes pueden seguir utilizando el servicio con normalidad. Para obtener más información, consulte [Cambio en la disponibilidad de AWS Mainframe Modernization](#).

El siguiente diagrama muestra cómo Prueba de aplicación de AWS funciona a un alto nivel.



El proceso consta de estos pasos:

1. Cree y gestione casos de prueba, que son la unidad más pequeña de acciones de prueba. Identifique los tipos de datos que mejor representan la equivalencia funcional entre los sistemas de origen y destino.
2. Defina la configuración del entorno de prueba especificando CloudFormation plantillas y atributos adicionales.
3. Cree conjuntos de pruebas, que son colecciones de casos de prueba.
4. Cargue y reproduzca conjuntos de datos: capture los conjuntos de datos de entrada y salida en la computadora central, cárguelos y, a continuación AWS, reproduzca el escenario de prueba en el sistema de destino.

5. Compare los conjuntos de datos de origen y destino. Prueba de aplicación de AWS compara automáticamente los conjuntos de datos de salida de los sistemas de origen y destino. Revíselos y evalúelos para identificar las discrepancias.

Para obtener más información, consulte la documentación de [AWS Mainframe Modernization](#).

Transición

En la modernización del mainframe, uno de los desafíos más importantes es minimizar el tiempo de inactividad y el riesgo durante la transición a una nueva plataforma. La estrategia blue/green de implementación ofrece un enfoque potente y flexible para la migración del sistema.

La implementación azul/verde es una técnica que reduce el tiempo de inactividad y el riesgo al ejecutar dos entornos de producción idénticos, denominados azul y verde. Así es como funciona en el contexto de la modernización del mainframe:

- Entorno azul: este es su sistema de mainframe actual que gestiona todo el tráfico de producción.
- Entorno ecológico: esta es su plataforma nueva y modernizada AWS que está lista para tomar el relevo.

La estrategia blue/green de transición incluye los siguientes pasos: aprovisionar, poner en marcha, revertir en caso de que surjan problemas y concluir.

Provisión

En esta etapa, se aprovisiona el nuevo entorno (ecológico) AWS siguiendo estos pasos:

1. Cambie la plataforma del entorno: la zona alojada de [Route 53](#) debe contener un [registro DNS](#) que apunte al entorno de mainframe (azul).
2. Verifique la conectividad: asegúrese de que haya una conexión adecuada entre sus administradores de transacciones Cuenta de AWS y los locales y Db2 for base de datos. z/OS
3. Realice pruebas de humo: utilice la dirección del balanceador de AWS carga para acceder al entorno rediseñado y realice pruebas de humo exhaustivas para verificar lo siguiente:
 - Están disponibles todas las cargas de trabajo esperadas.
 - 3270 transacciones se están procesando correctamente.

- Las interacciones de datos con Db2 for z/OS funcionan según lo esperado.

Transmite en directo

En esta etapa, se desplaza el tráfico a un entorno verde y se supervisan los cambios.

1. Use las políticas de enrutamiento de tráfico de Route 53 para desviar el tráfico:
 - Opción A: Puede cambiar el tráfico de una sola vez.
 - Opción B: Como alternativa, puede utilizar una distribución ponderada gradual.
2. Supervise y valide:
 - Observe de cerca el AWS entorno a medida que cambia el tráfico.
 - Compruebe el procesamiento de transacciones 3270.
 - Verifique la comunicación entre Db2. z/OS
 - Supervise los problemas de rendimiento.
 - Haga que los usuarios validen los resultados de las transacciones.

Reversión

Si surgen problemas, puede actualizar rápidamente Route 53 para redirigir el tráfico al entorno de mainframe local (azul).

Debe investigar y resolver los problemas antes de intentar realizar otra transición.

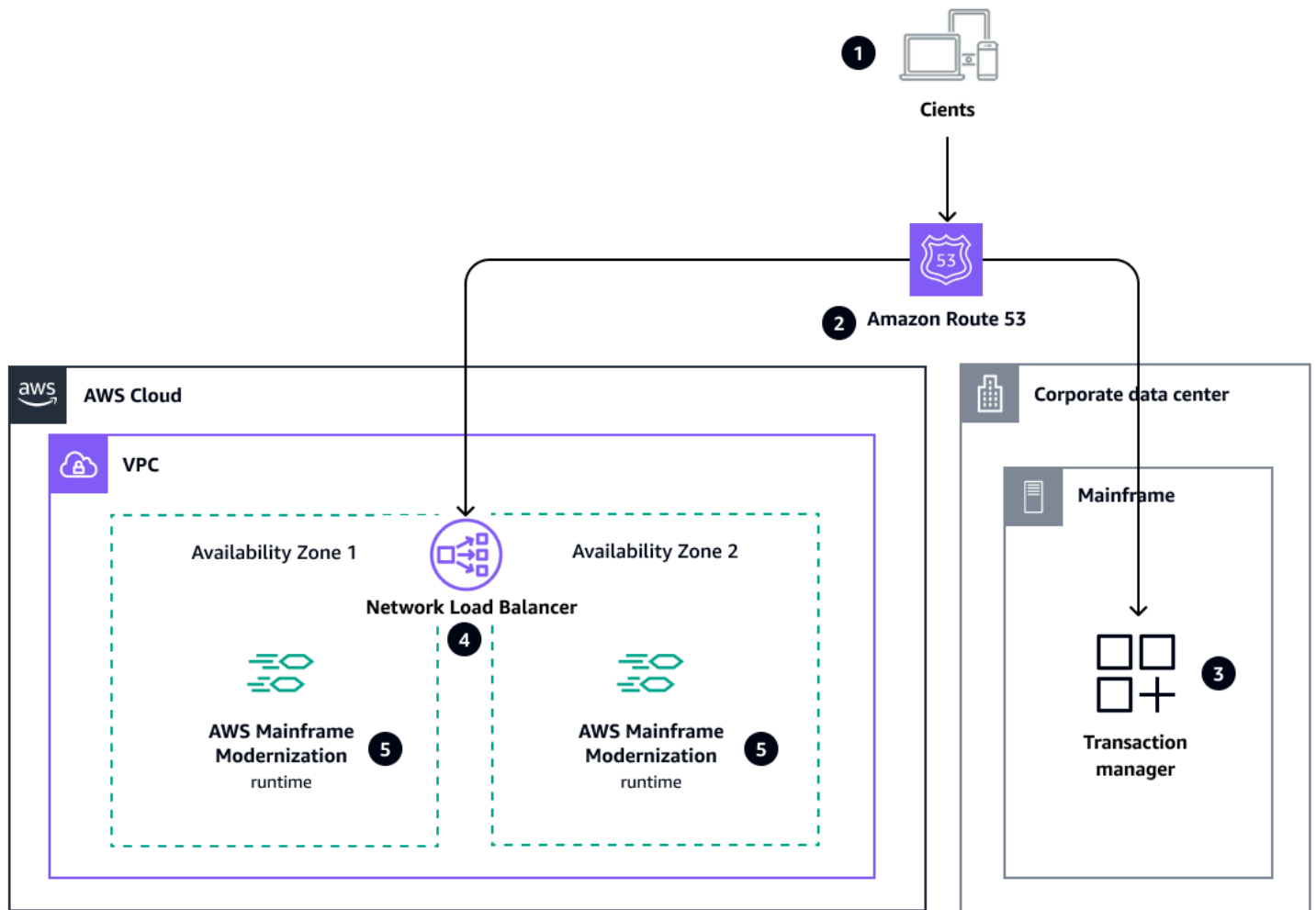
Concluir

Una vez que haya supervisado el tráfico y haya validado que su entorno ecológico funciona correctamente, podrá aumentar gradualmente el tráfico de aplicaciones a AWS.

Tras un período estable, puede dismantelar el entorno de transacciones del mainframe (azul) y mantener la z/OS base de datos Db2 for en las instalaciones.

Arquitectura

El siguiente diagrama ilustra el flujo de transición.



El proceso de transición consiste en lo siguiente:

1. Las aplicaciones cliente, las interfaces y los backends de las interfaces (BFFs) envían transacciones al nombre de dominio de Route 53.
2. Route 53 enruta la conexión al administrador de transacciones del mainframe o al Network Load Balancer, según la política de enrutamiento definida.
3. El administrador de transacciones procesa las transacciones que se envían al mainframe.
4. El Network Load Balancer distribuye las transacciones a los entornos de replataforma disponibles para su procesamiento.
5. Los entornos de AWS Mainframe Modernization replataforma procesan las solicitudes.

Prácticas recomendadas

En esta sección, se describe un conjunto de prácticas recomendadas para abordar los principales desafíos que supone la conversión de las cargas de trabajo del mainframe a entornos de nube y, al mismo tiempo, mantener la base de datos en Db2 for z/OS.

Latencia de la red

Para predecir con precisión el impacto que tendrá en la latencia la separación de la aplicación de la base de datos de Db2 durante un proceso de cambio de plataforma, le recomendamos que lleve a cabo una evaluación exhaustiva del número de llamadas a Db2, tanto para las transacciones como para los procesos por lotes. Esta evaluación debe realizarse mediante datos de rastreo e incluir los siguientes pasos:

- **Recopile datos de rastreo:** Recopile trazas detalladas de transacciones representativas y trabajos por lotes, y asegúrese de que las trazas capturen todas las interacciones de Db2, incluidas las entradas y salidas.
- **Analice los datos de rastreo:** cuente el número de entradas y salidas de Db2 para cada transacción y trabajo por lotes, y calcule el número promedio de interacciones de Db2 por transacción y proceso por lotes.
- **Mida los tiempos de respuesta actuales:** compruebe si el acceso a Db2 cumple con el acuerdo de nivel de servicio (SLA) de su aplicación.
- **Calcule la latencia de la red:** determine la latencia de red esperada entre la aplicación con la nueva plataforma y la base de datos de Db2. Tenga en cuenta factores como la distancia física, la infraestructura de red y los posibles cuellos de botella.
- **Calcule el impacto potencial:** para cada transacción y proceso por lotes, multiplique el número de entradas y salidas de Db2 por la latencia de red estimada. Agregue este tiempo calculado a los tiempos de respuesta actuales para predecir el nuevo tiempo total de procesamiento.
- **Evalúe los resultados:** evalúe si el aumento de latencia previsto es aceptable para los requisitos de su empresa e identifique cualquier transacción o proceso que pueda requerir una optimización o un rediseño para mitigar los problemas de latencia.
- **Considere estrategias de mitigación:** explore opciones como la agrupación de conexiones, el almacenamiento en caché o la recuperación de datos por lotes para reducir el número de

interacciones individuales de Db2. Evalúe la posibilidad de acercar los datos a los que se accede con frecuencia al nivel de la aplicación.

Si sigue estos pasos, podrá tomar decisiones basadas en datos sobre la viabilidad de su estrategia de cambio de plataforma e identificar cualquier posible problema de rendimiento antes de que afecte a su entorno de producción. Este enfoque ayudará a garantizar una transición fluida y, al mismo tiempo, a mantener niveles de rendimiento aceptables para las aplicaciones que dependen de las bases de datos.

Seguridad

- Proteja la creación de su aplicación: utilice una subred privada en la nube privada virtual (VPC) para AWS CodeBuild ejecutarla y así garantizar el aislamiento y mejorar la seguridad. Implemente un contexto de confianza de Db2 desde la CodeBuild subred CIDR para garantizar el acceso a la base de datos durante el proceso de creación.
- Proteja su entorno de ejecución: utilice un contexto confiable de Db2 de la subred de ejecución CIDR para proteger las conexiones de bases de datos.
- Administre las credenciales de la base de datos de forma segura: implemente un programa de rotación de credenciales regular para minimizar el riesgo de acceso no autorizado. Guarde las credenciales de Db2 de forma segura en AWS Secrets Manager
- Establezca la seguridad de la red: Implemente reglas sólidas de segmentación de red y firewall para proteger los entornos de construcción y tiempo de ejecución. Utilice la combinación correcta de AWS Direct Connect y AWS Site-to-Site VPN para lograr el nivel de seguridad necesario.
- Aplique el cifrado: aplique el cifrado de los datos en tránsito entre su aplicación y Db2 for z/OS.

Gobernanza de aplicaciones

- Establezca una fuente de verdad: establezca la nueva administración de configuración de software (SCM), por ejemplo, GitHub como la única fuente de verdad para el código de la aplicación migrada. Esto garantiza la coherencia y elimina las discrepancias de versión entre los entornos de nube y mainframe durante el período de transición.
- Actualice el proceso de administración de cambios: actualice el proceso de administración de cambios para tener en cuenta las modificaciones del código en este nuevo paradigma de doble entorno. Este proceso debe incluir:

- Flujos de trabajo de aprobación claros para cambios de código.
- Revisiones obligatorias del código antes de fusionarlo en la rama principal.
- Procedimientos de implementación sincronizados para garantizar que ambos entornos reciban actualizaciones simultáneamente.
- Mecanismos de reversión en caso de problemas en cualquiera de los entornos.

Elasticidad

La elasticidad de la computación en nube introduce un cambio de paradigma que altera significativamente la estructura de costos y la administración de recursos del mainframe. A diferencia del entorno de mainframe tradicional, que tiene una capacidad fija y modelos de precios basados en los picos de consumo, las plataformas en la nube ofrecen una escalabilidad dinámica y un pay-as-you-go enfoque que puede suponer un importante ahorro de costes y una mejora de la eficiencia operativa.

En un entorno de nube, las organizaciones pueden aumentar o reducir sus recursos informáticos en tiempo real en función de la demanda real, lo que elimina la necesidad de aprovisionamiento excesivo para adaptarse a los picos de carga. Esta elasticidad permite a las empresas pagar solo por los recursos que consumen, en lugar de invertir en costosas licencias de hardware y software para hacer frente a los picos ocasionales de uso.

Para obtener más información sobre cómo funcionan los precios AWS, consulte [AWS Precios](#).

Pasos siguientes

La modernización del mainframe es una iniciativa compleja y crítica que exige conocimientos especializados y soluciones avanzadas. Puede acelerar el proceso de modernización y lograr resultados empresariales más rápidos mediante [asociaciones estratégicas](#) que le ayudarán en las siguientes tareas:

- **Evalúe y priorice:** revise sus aplicaciones de mainframe e identifique cuáles son adecuadas para el cambio de plataforma y, al mismo tiempo, mantenga la base de datos en Db2 for z/OS. Tenga en cuenta factores como la complejidad, la importancia para el negocio y el posible retorno de la inversión (ROI).
- **Desarrolle una estrategia de migración:** cree un plan detallado para cambiar la plataforma de las aplicaciones seleccionadas, incluidos los plazos, la asignación de recursos y las estrategias de mitigación de riesgos.
- **Evalúe las herramientas y tecnologías:** investigue y seleccione las herramientas y tecnologías adecuadas para facilitar el proceso de cambio de plataforma, como las plataformas de modernización de aplicaciones o las herramientas de conversión de código.
- **Póngase en contacto con expertos:** considere la posibilidad de asociarse con especialistas en la modernización de mainframes o con firmas consultoras que tengan experiencia en proyectos de cambio de plataforma.
- **Prueba de concepto:** comience con una prueba de concepto a pequeña escala para validar su enfoque e identificar los posibles desafíos antes de ampliarlo a aplicaciones más grandes.
- **Pruebas y validación:** desarrolle una estrategia de pruebas integral para garantizar que sus aplicaciones reconfiguradas funcionen correctamente y mantengan la integridad de los datos con su base de datos de Db2 para base de datos existente. z/OS
- **Formación y transferencia de conocimientos:** prepare a su equipo para el nuevo entorno proporcionando formación sobre las aplicaciones reconfiguradas y cualquier herramienta o tecnología nueva que se introduzca.
- **Implementación gradual:** considere la posibilidad de adoptar un enfoque gradual para el cambio de plataforma, en el que se migren las aplicaciones de forma gradual y, al mismo tiempo, se supervise el rendimiento y se aborden los problemas que puedan surgir.
- **Optimización continua:** después de cambiar de plataforma, supervise y optimice continuamente el rendimiento de sus aplicaciones y sus interacciones con la base de datos de Db2 for para garantizar el éxito a largo plazo. z/OS

- **Modernice a su ritmo:** ahora que la carga de trabajo se está ejecutando AWS y ya está aprovechando las ventajas de la nube, comience a planificar la fase de rediseño de su modernización.

Recursos

Para obtener más información sobre la migración y modernización de mainframes, consulte los siguientes recursos.

AWS documentación

- [Configuración de Amazon Route 53 como servicio DNS](#)
- [Enrutar el tráfico a un balanceador de cargas ELB](#)
- [Enrutamiento ponderado](#)
- [Cambiar la plataforma de las aplicaciones con Rocket Software](#)

Referencias de Rocket Software

- [Interfaz de llamada externa \(ECI\) de Micro Focus](#)
- [Servicios web CICS](#)

Referencias de IBM

- [Contextos confiables](#) (IBM Db2 para z/OS documentación)

Tools (Herramientas)

- [Servidor Rocket Enterprise](#)

AWS Patrones y guías de orientación prescriptiva

- [Cree programas COBOL Db2 utilizando y AWS Mainframe ModernizationAWS CodeBuild](#)
- [DevOps para AWS Mainframe Modernization](#)
- [Modernización del mainframe: patrones de desacoplamiento para la migración del código de la aplicación](#)

- [Proteja y agilice el acceso de los usuarios a una base de datos de federación de Db2 mediante el AWS uso de contextos confiables](#)

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
Publicación inicial	—	7 de mayo 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 CCoE](#) 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
- Reinención: 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 alojarse en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a otras 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

I

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 (IIoT)

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 en la 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 dismantelar 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 versión original de inglés, prevalecerá la versión en inglés.