



La prueba de CI/CD fuego: ¿su proyecto está completamente integrado por CI/CD?

AWS Guía prescriptiva



AWS Guía prescriptiva: La prueba de CI/CD fuego: ¿su proyecto está completamente integrado por CI/CD?

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
Objetivos	1
Descripción de CI/CD	3
Acerca de la integración continua	4
Acerca de la entrega continua	4
Tests	5
Métricas	6
Diferencias de los procesos de CI/CD	8
Enfoque de Gitflow	8
Enfoque basado en enlaces troncales	10
Integridad del entorno	11
Versiones	12
Seguridad	13
Prueba de fuego para canalizaciones de CI/CD	15
Prácticas recomendadas	18
Preguntas frecuentes	20
¿Cuáles son algunos de los indicadores clave de que mi proceso de implementación no es una canalización de CI/CD completa?	20
¿Qué sucede si quiero utilizar un CI/CD proceso completo, pero aun así quiero programar el lanzamiento de determinadas funciones para momentos específicos?	20
¿Qué sucede si algunos pasos de mi proceso de implementación no se pueden automatizar?	20
¿Qué sucede si mi personal técnico se siente más cómodo con los flujos de trabajo tradicionales que con un CI/CD proceso completo?	21
¿Qué ocurre si mis entornos están en varias cuentas? ¿Puedo seguir utilizando un CI/CD proceso completo?	21
Pasos a seguir a continuación	22
Recursos	23
AWS documentación y referencias	23
Herramientas y servicios	23
Historial de documentos	24
Glosario	25
#	25
A	26

B	29
C	31
D	34
E	39
F	41
G	43
H	44
I	45
L	48
M	49
O	54
P	56
Q	59
R	60
S	63
T	67
U	68
V	69
W	69
Z	71
.....	lxxii

La prueba CI/CD de fuego: ¿su canalización es completamente CI/CD?

Steven Guggenheimer y Ananya Koduri, Amazon Web Services (AWS)

agosto de 2023 ([historial de documentos](#))

¿La canalización está automatizada? Es una pregunta sencilla, pero muchas organizaciones abordan la respuesta de forma demasiado sencilla. La respuesta es mucho más complicada que un sí o un no.

Las innovaciones tecnológicas se producen constantemente y, a veces, puede resultar difícil para las organizaciones mantenerse al día. ¿Esta novedad es una moda pasajera o es la próxima gran novedad? ¿Debo revisar mis prácticas actuales o debo esperar? A menudo, cuando queda claro que algo va a ser la próxima gran novedad, puede verse que se está intentando poner al día. La integración continua y la entrega continua (CI/CD) llegaron para quedarse, pero no siempre fue así. A muchas personas les llevó mucho tiempo convencerlas, mientras que algunas aún necesitan convencerse más.

CI/CD is the process of automating the source, build, test, staging, and production stages of the software release process, and it is commonly described as a pipeline. Today, the cost savings and speed of CI/CD las automatizaciones han convencido a la mayoría de las organizaciones de su valor. Sin embargo, la transición a este nuevo enfoque no es una tarea fácil. Debe asegurarse de que el personal tenga la formación adecuada, mejorar algunos recursos y, posteriormente, hacer pruebas y más pruebas. Hay mucho por hacer. En la mayoría de los casos, es conveniente que haga estos cambios de forma gradual para que la organización pueda adaptarse.

El propósito de este documento es definir lo que significa tener un CI/CD proceso completo. Proporciona una herramienta para evaluar los procesos propios y presenta un camino a seguir para los procesos que aún no están listos. Este camino que seguir rara vez es una conversión de la noche a la mañana. Estos procesos son complejos y dependen de muchos factores, como las habilidades actuales de los empleados y las demandas actuales de infraestructura. Le recomendamos que priorice y haga cambios pequeños e incrementales.

Objetivos

Los beneficios potenciales de implementar las recomendaciones de esta guía son los siguientes:

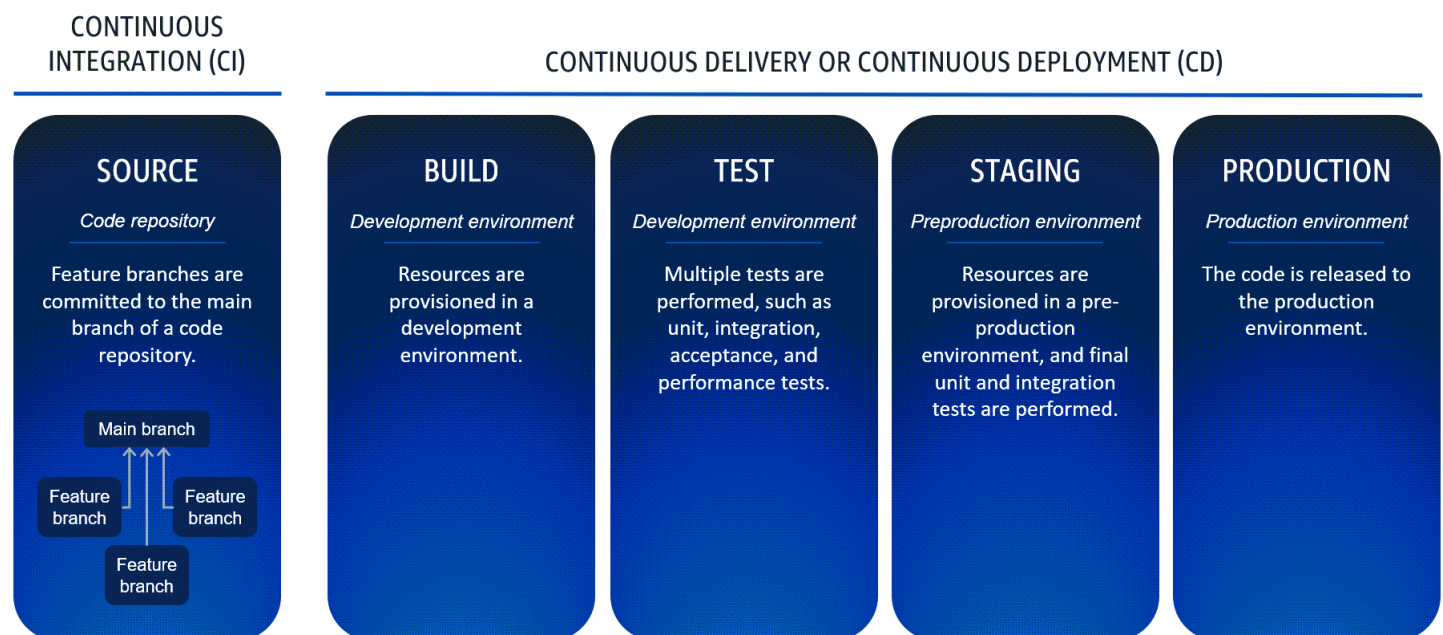
- **Eficiencia:** un proceso de CI/CD implementación completo puede reducir la complejidad, las cargas de trabajo y las innumerables horas dedicadas a la depuración, la realización de procesos manuales y el mantenimiento. Para obtener más información, consulte [Beneficios de la entrega continua](#). Según una entrada de [TechAhead blog](#), la implementación del CI/CD proceso puede suponer un ahorro estimado del 20% en tiempo, esfuerzo y recursos.
- **Reducción de costos:** según un [informe de Forbes Insight](#), “tres de cada cuatro ejecutivos están de acuerdo en que la cantidad de tiempo, dinero y recursos que se dedican al mantenimiento y la administración continuos, en comparación con el desarrollo de nuevos proyectos o nuevas iniciativas, afectan a la competitividad general de su organización”. Cuanto más corto sea el ciclo de desarrollo, mayores serán las probabilidades de que su organización pueda cumplir time-to-market objetivos ambiciosos y aprovechar las oportunidades adecuadas en el momento adecuado.
- **Velocidad:** por lo general, una CI/CD pipeline is able to release software changes to customers within a few hours. Especially in cases with quick fault isolations and small patch pushes, the CI/CD canalización completa ayuda a mejorar el tiempo medio de recuperación (MTTR). Para más información, consulte [Reducing MTTR](#).
- **Seguridad:** los CI/CD procesos completos también protegen el proceso de liberación al reducir los posibles puntos de entrada de los ataques y reducir el riesgo de errores humanos. Las mejoras de seguridad que se obtienen con los procesos totalmente CI/CD automatizados ayudan a evitar las costosas consecuencias de las filtraciones de datos, las interrupciones del servicio, etc.
- **Reducción del agotamiento:** los desarrolladores se sienten más satisfechos cuando pueden dedicar más tiempo a crear excelentes características y menos tiempo a sumergirse en un ciclo interminable de mantenimiento y depuración. Para las organizaciones, esto significa adquirir y retener a los mejores talentos durante periodos de tiempo más largos.
- **Código de calidad superior:** los desarrolladores publican el código en un repositorio compartido en pequeños lotes, lo que les permite realizar [pruebas paralelas](#) (entrada del BrowserStack blog). En lugar de trabajar de forma aislada, comparten sus compilaciones con el equipo frecuentemente y colaboran para identificar los errores críticos. Esto proporciona soporte a los desarrolladores, lo que ayuda a evitar que el código incorrecto llegue a producción. El soporte de compañeros desarrolladores contribuye a la creación de lanzamientos de alta calidad e impulsa el crecimiento de la organización.
- **Mantenimiento:** el mantenimiento y las actualizaciones son una parte crucial de la creación de un gran producto. Sin embargo, no desactive el sistema durante las horas de mayor tráfico. Puede utilizar una CI/CD canalización para realizar el mantenimiento durante las horas de bajo consumo, lo que minimiza el tiempo de inactividad y el impacto en el rendimiento.

Descripción de CI/CD

La integración y entrega continuas (CI/CD) es el proceso de automatización del ciclo de vida de los lanzamientos de software. En algunos casos, la entrada D también CI/CD puede significar despliegue. La diferencia entre la entrega continua y la implementación continua se produce cuando se publica un cambio en el entorno de producción. Con la entrega continua, se requiere aprobación manual antes de pasar cambios a producción. La implementación continua ofrece un flujo ininterrumpido en toda la canalización y no se requieren aprobaciones explícitas. Como en esta estrategia se analizan CI/CD conceptos generales, las recomendaciones y la información proporcionadas son aplicables tanto a los enfoques de entrega continua como de despliegue continuo.

CI/CD automatiza muchos o todos los procesos tradicionalmente requeridos para obtener nuevo código desde un commit hasta producción. Una CI/CD pipeline abarca las etapas de fuente, construcción, prueba, staging y producción. En cada etapa, las pipelines de CI/CD provisionan cualquier infraestructura que se necesita para desplegar o probar el código. Al utilizar un proceso de CI/CD, los equipos de desarrollo pueden realizar cambios en el código que, posteriormente, se prueban automáticamente y luego se implementan.

Repasemos el CI/CD proceso básico antes de analizar algunas de las formas en que puedes, consciente o inconscientemente, dejar de ser CI/CD. The following diagram shows the CI/CD etapas completas y actividades en cada etapa.



Acerca de la integración continua

La integración continua se produce en un repositorio de código, como un repositorio de Git en GitHub. Se trata una sola rama principal como origen de información para el código base y se crean ramas de corta duración para el desarrollo de características. Una rama de características se integra en la rama principal cuando lo tiene todo listo para implementar la característica en los entornos superiores. Las ramas de características nunca se implementan directamente en los entornos superiores. Para obtener más información, consulte la sección [Enfoque basado en enlaces troncales](#) de esta guía.

Proceso de integración continua

1. El desarrollador crea una nueva rama a partir de la rama principal.
2. El desarrollador hace cambios, compila y prueba de forma local.
3. Cuando los cambios están listos, el desarrollador crea una [solicitud](#) de cambios (GitHub documentación) con la sucursal principal como destino.
4. Se revisa el código.
5. Cuando se aprueba el código, se fusiona con la rama principal.

Acerca de la entrega continua

La entrega continua se produce en entornos aislados, como entornos de desarrollo y de producción. Las acciones que se producen en cada entorno pueden variar. Una de las primeras etapas se suele utilizar para llevar a cabo actualizaciones en la propia canalización antes de continuar. El resultado final de la implementación es que cada entorno se actualiza con los cambios más recientes. La cantidad de entornos de desarrollo para la creación y las pruebas también varía, pero le recomendamos que utilice al menos dos. En el proceso, cada entorno se actualiza por orden según su importancia y termina con el entorno más importante, el entorno de producción.

Proceso de entrega continua

La parte del proceso de entrega continua se inicia con la extracción del código de la rama principal del repositorio de origen y su traslado a la etapa de compilación. El documento de infraestructura como código (IaC) del repositorio describe las tareas que se llevan a cabo en cada etapa. Si bien el uso de un documento de IaC no es obligatorio, se recomienda encarecidamente utilizar un servicio o una herramienta de IaC, como [AWS CloudFormation](#) o [AWS Cloud Development Kit \(AWS CDK\)](#). Los pasos más comunes son:

1. Pruebas unitarias
2. Compilación de código
3. Aprovisionamiento de recursos
4. Pruebas de integración

Si se produce algún error o alguna prueba falla en alguna etapa de la canalización, la etapa actual vuelve a su estado anterior y la canalización termina. Los cambios posteriores deben comenzar en el repositorio de código y continuar con el CI/CD proceso completo.

Pruebas de CI/CD canalización

Los dos tipos de pruebas automatizadas a las que se suele hacer referencia en las canalizaciones de implementación son las pruebas unitarias y las pruebas de integración. Sin embargo, hay muchos tipos de pruebas que se pueden ejecutar en un código base y en el entorno de desarrollo. La [arquitectura de referencia para canalizaciones de implementación de AWS](#) define los siguientes tipos de pruebas:

- Prueba unitaria: estas pruebas compilan y ejecutan el código de la aplicación para comprobar que funcione según las expectativas. Simulan todas las dependencias externas que se utilizan en el código base. Entre los ejemplos de herramientas de pruebas unitarias [JUnit](#) se incluyen [Jest](#) y [pytest](#).
- Prueba de integración: estas pruebas verifican que la aplicación cumpla con los requisitos técnicos mediante la prueba en un entorno de prueba aprovisionado. Entre los ejemplos de herramientas de pruebas de integración se incluyen [Cucumber](#), [vRest NG](#), e [integ-tests](#) (para AWS CDK).
- Prueba de aceptación: estas pruebas verifican que la aplicación cumpla con los requisitos del usuario mediante la prueba en un entorno de prueba aprovisionado. Entre los ejemplos de herramientas de pruebas de aceptación se incluyen [Cypress](#) y [Selenium](#).
- Prueba sintética: estas pruebas se ejecutan de forma continua en segundo plano para generar tráfico y verificar que el sistema esté en buen estado. Algunos ejemplos de herramientas de prueba sintéticas son [Amazon CloudWatch Synthetics](#) y [Dynatrace Synthetic](#) Monitoring.
- Prueba de rendimiento: estas pruebas simulan la capacidad de producción. Determinan si la aplicación cumple con los requisitos de rendimiento y comparan las métricas con el rendimiento anterior. [Entre los ejemplos de herramientas de prueba de rendimiento se incluyen Apache JMeter, Locust y Gatling.](#)

- Prueba de resiliencia: también conocidas como pruebas de caos, estas pruebas inyectan errores en los entornos para identificar las áreas de riesgo. Los periodos en los que se producen los errores se comparan luego con los periodos sin errores. Entre los ejemplos de herramientas de pruebas de resiliencia se incluyen [AWS Fault Injection Service](#) y [Gremlin](#).
- Prueba estática de seguridad de aplicaciones (SAST): estas pruebas analizan el código en busca de infracciones de seguridad, como una [inyección de código SQL](#) o el [scripting entre sitios \(XSS\)](#). Algunos ejemplos de herramientas SAST incluyen [Amazon CodeGuru](#) y [Checkmarx](#). [SonarQube](#)
- Prueba dinámica de seguridad de aplicaciones: estas pruebas también se conocen como pruebas de penetración o pruebas de incursión. Identifican vulnerabilidades, como la inyección de código SQL o el XSS, en un entorno de prueba aprovisionado. [Algunos ejemplos de herramientas de DAST incluyen Zed Attack Proxy \(ZAP\) y HCL. AppScan](#) Para más información, consulte [Penetration Testing](#).

No todas las CI/CD canalizaciones completas ejecutan todas estas pruebas. Sin embargo, como mínimo, una canalización debería ejecutar pruebas unitarias y pruebas de SAST en el código base, así como pruebas de integración y aceptación en un entorno de pruebas.

Métricas de las canalizaciones CI/CD

De acuerdo con la [arquitectura de referencia de las canalizaciones de AWS despliegue](#), deberías, como mínimo, realizar un seguimiento de las cuatro métricas siguientes para las CI/CD canalizaciones:

- Plazo de entrega: el promedio de tiempo que tarda una sola confirmación en llegar a la fase de producción. Le recomendamos que se centre en un plazo de entre 1 hora y 1 día, según corresponda a su caso de uso.
- Frecuencia de implementación: el número de implementaciones de producción en un periodo de tiempo determinado. Le recomendamos segmentar las frecuencias de implementación entre varias veces al día y dos veces por semana, según corresponda a su caso de uso.
- Tiempo medio entre errores (MTBD): el promedio de tiempo entre el inicio de una canalización correcta y el inicio de una canalización con errores. Le recomendamos que opte por un MTBD lo más alto posible. Para más información, consulte [Increasing MTBF](#).
- Tiempo medio de recuperación (MTTR): el promedio de tiempo que transcurre entre el inicio de una canalización con errores y el inicio de la siguiente canalización correcta. Le recomendamos que opte por un MTTR lo más bajo posible. Para más información, consulte [Reducing MTTR](#).

Estas métricas ayudan a los equipos a hacer un seguimiento de su progreso para obtener una CI/CD completa. Los equipos deben mantener conversaciones abiertas con las partes interesadas de la organización sobre cuáles deberían ser los objetivos óptimos. Las situaciones y necesidades varían mucho de una organización a otra, e incluso de un equipo a otro.

Es muy importante recordar que los cambios rápidos y drásticos suelen aumentar el riesgo de que surjan problemas. Establezca objetivos para lograr mejoras pequeñas e incrementales. Un plazo de entrega óptimo habitual para CI/CD canalizaciones completas es inferior a 3 horas. Un equipo que comience con un plazo de entrega de 5,2 días debería tener como objetivo una reducción de un día cada pocas semanas. Cuando este equipo cumpla un plazo de entrega de un día o menos, podrá permanecer allí durante varios meses y adoptar un plazo de entrega más agresivo solo si el equipo y las partes interesadas de la organización lo consideran necesario.

Cuán completamente diferentes son los CI/CD procesos

Las canalizaciones de CI/CD utilizan un flujo de trabajo moderno basado en enlaces troncales, en el que los desarrolladores combinan las actualizaciones pequeñas y frecuentes en una rama principal (o troncal) que se crea y prueba a través de la parte de CD de la canalización. CI/CD Este flujo de trabajo ha sustituido al flujo de trabajo de Gitflow, en el que las ramas de desarrollo y lanzamiento están separadas por una programación de lanzamiento. En muchas organizaciones, Gitflow sigue siendo un método popular de control de versiones e implementación. Sin embargo, ahora se considera heredado y su integración en una canalización puede resultar difícil. CI/CD

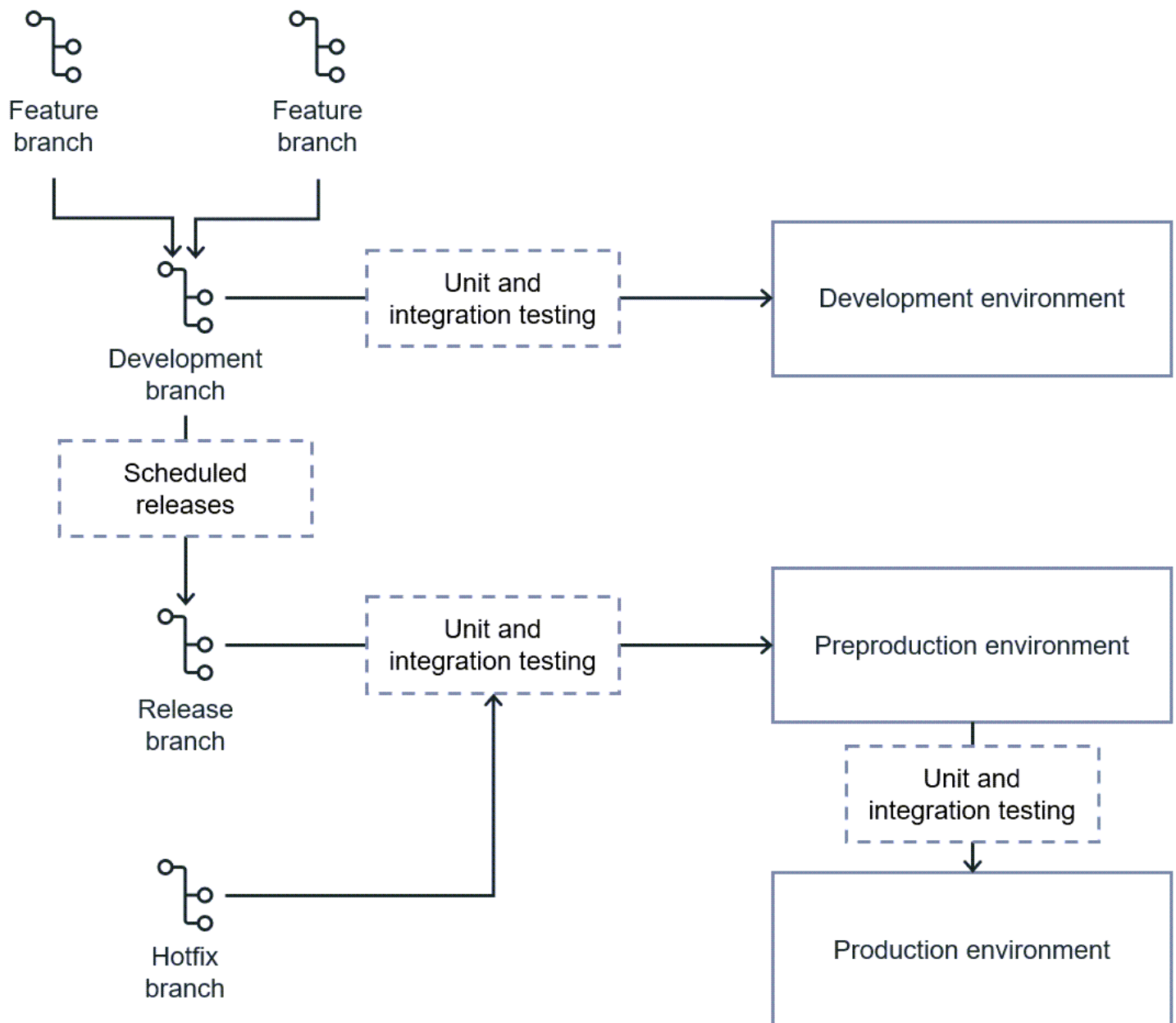
Para muchas organizaciones, la transición de un flujo de trabajo de Gitflow a un flujo de trabajo basado en enlaces troncales no se ha completado, y el resultado es que se quedan estancadas en algún momento y nunca migran completamente a la CI/CD. De alguna manera, las canalizaciones acaban aferrándose a algunos remanentes del flujo de trabajo heredado, atrapados en un estado de transición entre el pasado y el presente. Revise las diferencias en los flujos de trabajo de Git y obtenga información sobre cómo el uso de un flujo de trabajo heredado puede afectar a lo siguiente:

- [Integridad del entorno](#)
- [Versiones](#)
- [Seguridad](#)

Para facilitar la identificación de los remanentes de un flujo de trabajo de Git heredado en una configuración moderna, comparemos [Gitflow](#) con el enfoque moderno basado en [enlaces troncales](#).

Enfoque de Gitflow

En la siguiente imagen se muestra un flujo de trabajo de Gitflow. El enfoque de Gitflow usa múltiples ramas para hacer un seguimiento de varias versiones diferentes del código al mismo tiempo. Se programan los lanzamientos de actualizaciones de una aplicación para algún momento en el futuro mientras los desarrolladores siguen trabajando en la versión actual del código. Los repositorios basados en enlaces troncales pueden usar marcadores de características para lograrlo, pero está integrado en Gitflow de forma predeterminada.



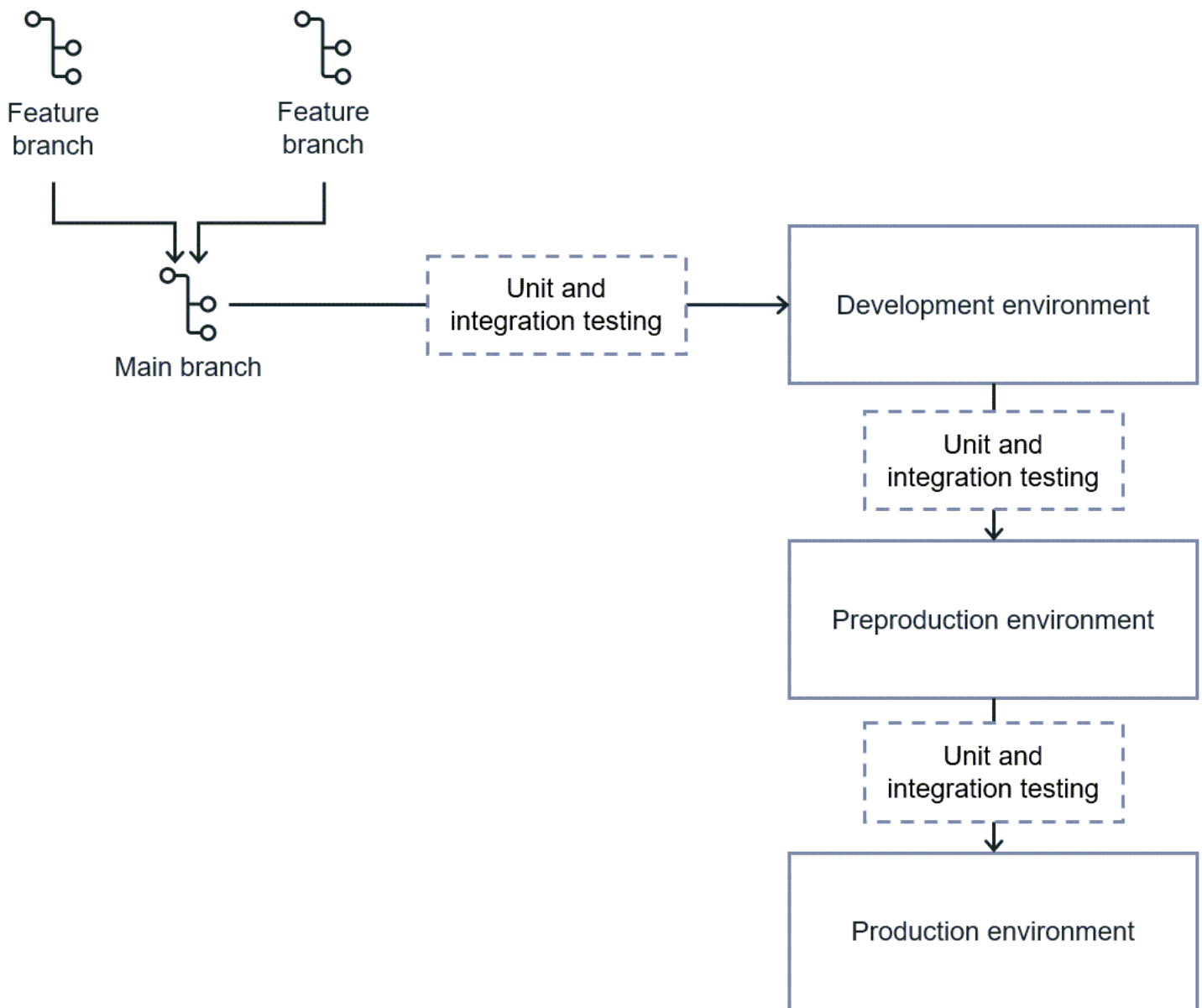
Uno de los resultados del enfoque de Gitflow es que los entornos de las aplicaciones no suelen estar sincronizados. En una implementación estándar de Gitflow, los entornos de desarrollo reflejan el estado actual del código, mientras que los entornos de preproducción y producción permanecen inmóviles en función del estado del código base desde la versión más reciente.

Esto complica las cosas cuando aparece un defecto en el entorno de producción, ya que el código base con el que trabajan los desarrolladores no se puede fusionar con el entorno de producción sin exponer características que no se hayan lanzado. La forma en que Gitflow maneja esta situación es mediante una revisión. Se crea una rama de revisiones a partir de la rama de lanzamiento y, a

continuación, se implementa directamente en los entornos posteriores. Luego, la rama de revisiones se fusiona con la rama de desarrollo para mantener el código actualizado.

Enfoque basado en enlaces troncales

En la siguiente imagen se muestra un flujo de trabajo basado en enlaces troncales. En un flujo de trabajo basado en enlaces troncales, los desarrolladores crean y prueban características de forma local en una rama de características 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. Las pruebas unitarias y de integración se llevan a cabo entre cada entorno.



Con este flujo de trabajo, todos los entornos funcionan con el mismo código base. No es necesaria ninguna rama de revisiones para los entornos superiores, ya que se pueden implementar cambios en la rama principal sin exponer características que no se hayan lanzado. Siempre se supone que la rama principal es estable, no tiene defectos y está lista para su lanzamiento. Esto le ayuda a integrarlo como fuente de una CI/CD canalización, que puede probar e implementar automáticamente su base de código en todos los entornos de la canalización.

Beneficios de un enfoque basado en enlaces troncales para la integridad del entorno

Como muchos desarrolladores saben, un cambio en el código a veces puede provocar un [efecto mariposa](#) (artículo de American Scientist), en el que una pequeña desviación aparentemente no relacionada desencadena una reacción en cadena que provoca resultados inesperados. A continuación, los desarrolladores deben investigar a fondo para descubrir la causa raíz.

Cuando los científicos llevan a cabo un experimento, separan a los sujetos de prueba en dos grupos: el grupo experimental y el grupo de control. La intención es hacer que el grupo experimental y el grupo de control sean completamente idénticos, excepto en lo que respecta a lo que se prueba en el experimento. Cuando ocurre algo en el grupo experimental que no ocurre en el grupo de control, la única causa puede ser lo que se está probando.

Piense en los cambios de una implementación como el grupo experimental y piense en cada entorno como grupos de control independientes. Los resultados de las pruebas en un entorno inferior solo son fiables cuando los controles son los mismos que en un entorno superior. Cuanto más se desvíen los entornos, mayor será la probabilidad de descubrir defectos en los entornos superiores. En otras palabras, si los cambios en el código van a fallar en producción, preferimos que fallen antes en la versión beta para que nunca lleguen a la fase de producción. Por este motivo, se debe hacer todo lo posible para mantener sincronizados todos los entornos, desde el entorno de prueba más bajo hasta el propio entorno de producción. Esto se denomina integridad del entorno.

El objetivo de todo CI/CD proceso completo es descubrir los problemas lo antes posible. Preservar la integridad del entorno mediante un enfoque basado en enlaces troncales puede eliminar prácticamente la necesidad de revisiones. En un flujo de trabajo basado en enlaces troncales, es poco frecuente que un problema aparezca por primera vez en el entorno de producción.

En un enfoque de Gitflow, una vez que una revisión se implementa directamente en los entornos superiores, se agrega a la rama de desarrollo. Esto preserva la solución para futuras versiones. Sin embargo, la revisión se desarrolló y probó directamente a partir del estado actual de la aplicación.

Incluso si la revisión funciona perfectamente en producción, existe la posibilidad de que surjan problemas cuando interactúe con las características más nuevas de la rama de desarrollo. Como no suele ser deseable implementar una revisión para una revisión, esto hace que los desarrolladores dediquen más tiempo a intentar adaptar la revisión al entorno de desarrollo. En muchos casos, esto puede generar una importante deuda técnica y reducir la estabilidad general del entorno de desarrollo.

Cuando se produce un error en un entorno, se revierten todos los cambios para que el entorno vuelva a su estado anterior. Cualquier cambio en el código base debería volver a iniciar la canalización desde la primera etapa. Cuando surge un problema en el entorno de producción, la solución también debería pasar por todo el proceso. El tiempo adicional que se tarda en pasar por los entornos más bajos suele ser insignificante en comparación con los problemas que se evitan con este enfoque. Como el único propósito de los entornos inferiores es detectar los errores antes de que lleguen a producción, eludir estos entornos mediante un enfoque de Gitflow es un riesgo ineficiente e innecesario.

Beneficios de lanzamiento de un enfoque basado en enlaces troncales

Una de las cosas que suele hacer necesaria una revisión es que, en un flujo de trabajo heredado, el estado de la aplicación en la que están trabajando los desarrolladores puede contener varias características inéditas que aún no están en producción. El entorno de producción y el entorno de desarrollo solo se sincronizan cuando se produce un lanzamiento programado y, de inmediato, comienzan a diferir de nuevo hasta la siguiente versión programada.

Es posible programar los lanzamientos dentro de un proceso completo CI/CD . Puede retrasar el lanzamiento del código a producción mediante el uso de marcadores de características. Sin embargo, un CI/CD proceso completo permite una mayor flexibilidad al hacer innecesarias las publicaciones programadas. Después de todo, continua es una palabra clave en la CI/CD, y eso sugiere que los cambios se publican a medida que están listos. Evite mantener un entorno de lanzamiento independiente que casi nunca esté sincronizado con los entornos de prueba inferiores.

Si una canalización no es de CI/CD completa, la divergencia entre los entornos superior e inferior suele producirse por rama. Los desarrolladores trabajan en una rama de desarrollo y mantienen una rama de publicación independiente que solo se actualiza cuando llega el momento de una publicación programada. A medida que la rama de lanzamiento y de desarrollo divergen, pueden surgir otras complicaciones.

Además de que los entornos no están sincronizados, a medida que los desarrolladores trabajan en la rama de desarrollo y se van acostumbrando a un estado de las aplicaciones muy superior al de producción, deben reajustarse al estado de producción cada vez que surge un problema. El estado de la rama de desarrollo podría consistir en muchas características antes de la producción. Cuando los desarrolladores trabajan en esa rama todos los días, es difícil recordar qué se ha lanzado y qué no se ha lanzado a producción. Esto agrega el riesgo de que se ingresen nuevos errores mientras se corrigen otros. El resultado es un ciclo aparentemente interminable de correcciones que prolonga los plazos y retrasa el lanzamiento de las características durante semanas, meses o incluso años.

Beneficios de seguridad de un enfoque basado en enlaces troncales

Un CI/CD proceso completo proporciona un enfoque de implementación totalmente automatizado y con una única fuente de información. La canalización tiene un único punto de entrada. Las actualizaciones de software entran en la canalización desde el principio y se transmiten tal cual de un entorno a otro. Si se detecta un problema en cualquier fase del proceso, los cambios en el código que lo solucionen deben pasar por el mismo proceso y comenzar en la primera fase. Al reducir los puntos de entrada en una canalización, también se reducen las posibles formas de ingresar vulnerabilidades en la canalización.

Además, como el punto de entrada es el punto más alejado posible del entorno de producción, se reduce drásticamente la probabilidad de que las vulnerabilidades lleguen a la producción. Si se implementa un proceso de aprobación manual en un proceso íntegramente relacionado con la CI/CD, aún puede dejar de tomar decisiones sobre si los cambios se deben pasar o no al siguiente entorno. La persona responsable de la toma las decisiones no es necesariamente la misma que implementa los cambios. Esto separa las responsabilidades de quien implementa los cambios de código y de quien los aprueba. También hace que sea más factible que un líder de la organización menos técnico tenga el rol de aprobador.

Por último, el punto de entrada único permite limitar el acceso de escritura a la consola de la interfaz de usuario del entorno de producción a pocos usuarios o incluso a ninguno. Al reducir el número de usuarios que pueden hacer cambios manuales en la consola, se reduce el riesgo de que se produzcan problemas de seguridad. La capacidad de gestionar manualmente la consola en el entorno de producción es mucho más necesaria en los flujos de trabajo tradicionales que en un enfoque CI/CD automatizado. Estos cambios manuales son más difíciles de rastrear, revisar y probar. Por lo general, se hacen para ahorrar tiempo, pero, a la larga, agregan una importante deuda técnica al proyecto.

Los problemas de seguridad de la consola no se deben necesariamente a personas malintencionadas. Muchos de los problemas que se producen en la consola son accidentales. Las exposiciones de la seguridad accidentales son muy comunes y han hecho que surja el modelo de seguridad de confianza cero. Este modelo postula, en parte, que los accidentes de seguridad son menos probables cuando incluso el personal interno tiene el menor acceso posible, lo que también se conoce como permisos con privilegios mínimos. Preservar la integridad del entorno de producción mediante la restricción de todos los procesos a una canalización automatizada prácticamente elimina el riesgo de problemas de seguridad relacionados con la consola.

Prueba de fuego para canalizaciones de CI/CD

En química, el papel tornasol es una tira delgada de papel tratada con un tinte rojo o azul especial que se usa para determinar la acidez de una sustancia. Un ácido convierte el papel tornasol azul en rojo, una sustancia básica vuelve rojo el papel tornasol azul y las sustancias neutras no afectan en absoluto al color del papel.

La forma en que el papel tornasol determina la acidez es la medición del nivel de pH de una sustancia. Si un nivel de pH es superior a 8, es ácido; si está por debajo de 5, es básico; y si está entre 5 y 8, es neutro. Del mismo modo, la [prueba de fuego de CI/CD](#) lo ayuda a medir el nivel de CI/CD de la canalización.

Cómo probar si la canalización es de CI/CD completa

1. Comience con una puntuación de 0.
2. Responda a cada una de las siguientes preguntas y agregue 1 a la puntuación por cada respuesta afirmativa:
 - ¿Cada uno de los repositorios tiene exactamente una rama principal que se utiliza para implementar en los entornos?
 - ¿Enviamos el código a la rama principal con frecuencia y evitamos que las ramas de características se ejecuten durante mucho tiempo?
 - ¿La canalización tiene un único punto de entrada? En otras palabras, ¿la canalización extrae el código de cada repositorio exactamente una vez?
 - ¿Tenemos más de un entorno de implementación?
 - Cuando la canalización no está en ejecución, ¿los entornos superior e inferior suelen estar sincronizados?
 - ¿Ejecutamos pruebas en el código antes de implementarlo?
 - ¿Ejecutamos pruebas en un entorno antes de pasar al siguiente entorno?
 - ¿La canalización se revierte por completo y se cierra después de un error?
 - ¿La canalización se reinicia desde el primer paso al recuperarse de un error?
 - ¿Seguimos el mismo proceso para corregir errores en la producción que para lanzar características a producción?
 - ¿Usamos algún tipo de plantilla de infraestructura como código (IaC) para implementar código?

3. Responda a cada una de las siguientes preguntas y agregue 1 a la puntuación por cada respuesta negativa:
- ¿Alguna vez implementamos directamente en un entorno de implementación desde ramas distintas de la principal?
 - ¿Alguna vez implementamos directamente desde cualquier rama a un entorno superior o de producción?
 - ¿Solemos encontrar errores en los entornos superiores que no estaban presentes en los entornos inferiores?
 - ¿Alguna vez eludimos los entornos inferiores durante una implementación?
 - ¿Tenemos que esperar hasta la fecha de lanzamiento programada para implementarla en producción?
 - ¿Llevamos a cabo actualizaciones periódicas en la consola del entorno de producción?
 - ¿Hay algún paso de implementación manual que deba hacerse en la consola del entorno de producción para completar la implementación?
 - ¿Hay más de una persona con acceso de escritura al entorno de producción?
 - ¿Hay más de cinco personas con acceso de escritura al entorno de producción?
4. Divida la puntuación entre 2. Esta es la puntuación de CI/CD de la canalización.
5. Compare la puntuación de CI/CD de la canalización con la siguiente tabla para determinar el nivel de CI/CD de la canalización.

Puntuación de CI/CD	Nivel de CI/CD
9,5 o más	CI/CD completa
Entre 8 y 9	Principalmente CI/CD
5–7	Neutral
Por debajo de 5	No es CI/CD

Si obtuvo una puntuación inferior a 8, le recomendamos que se fije un objetivo para avanzar gradualmente hacia el siguiente nivel. Cuando se logra ese objetivo, las partes interesadas del producto deberán evaluar si es necesario establecer un nuevo objetivo y cuándo. La intención de este ejercicio no es necesariamente abogar por un cambio en la canalización, sino más bien dar a

conocer cómo es un proceso de implementación de CI/CD completa y cuál es la posición actual de las canalizaciones en ese espectro.

Prácticas recomendadas para canalizaciones de CI/CD

A continuación se enumeran las prácticas recomendadas para canalizaciones de CI/CD completas:

- **Proteja el entorno de producción:** como es posible lograr prácticamente todo lo necesario para el mantenimiento de cuentas y entornos mediante IaC, es importante esforzarse al máximo para proteger el entorno de producción mediante la limitación del acceso programático y la consola. Recomendamos limitar el acceso a pocos usuarios, o incluso a ninguno. Al implementar IaC a través de AWS CloudFormation, el usuario necesita permisos limitados. La mayoría de los permisos se asignan al servicio de CloudFormation a través de un rol de servicio. Para más información, consulte [Rol de servicio](#) en la documentación de CloudFormation e [Implementing policies for least-privilege permissions for AWS CloudFormation](#).
- **Cree cuentas independientes para cada entorno:** al dedicar una cuenta independiente a cada entorno, puede simplificar el proceso de implementación y crear controles de acceso detallados por cuenta. Cuando varios entornos comparten recursos, se reduce la integridad del entorno como unidad aislada. Lo mejor es mantener los entornos sincronizados y diferenciados. Esto es aún más importante para el entorno de producción, ya que todo lo que contiene esa cuenta debe tratarse como un recurso de producción.
- **Limite la información de identificación personal (PII) al entorno de producción:** tanto por motivos de seguridad como de protección contra los riesgos de responsabilidad, proteja la PII en la medida de lo posible. Cuando sea posible en entornos más bajos, utilice datos anonimizados o de muestra en lugar de copiar información potencialmente confidencial del entorno de producción.
- **Revise el código en los repositorios:** un proceso de CI/CD completo reduce los puntos de entrada de una canalización a un solo punto, que debe estar protegido. Por este motivo, se recomienda solicitar varias revisiones del código antes de fusionar las ramas de características con la rama principal. Estas revisiones de código las puede llevar a cabo cualquier miembro cualificado del equipo, pero debería revisarlas al menos un miembro sénior. El revisor debe probar rigurosamente el código. Al fin y al cabo, la mejor manera de solucionar los problemas de una canalización es evitar ingresarlos en ella. Además, es importante resolver todos los comentarios que haga cualquier revisor antes de la fusión. Esta resolución podría ser simplemente una explicación de por qué no es necesario hacer cambios, pero abordar todos los comentarios es una medida adicional importante para evitar que se ingresen problemas en el proceso.
- **Haga fusiones pequeñas y frecuentes:** para aprovechar al máximo la integración continua, es una buena idea ingresar también los cambios locales de forma continua en la canalización. Al fin y al

cabo, es mucho más beneficioso que los entornos de desarrollo se mantengan sincronizados si los entornos locales también se mantienen al día.

Para más información sobre las prácticas recomendadas para canalizaciones de CI/CD, consulte [Resumen de las prácticas recomendadas](#) en Práctica de integración y entrega continuas en AWS.

Preguntas frecuentes

¿Cuáles son algunos de los indicadores clave de que mi proceso de implementación no es una canalización de CI/CD completa?

El indicador más común es cuando hay varias ramas de repositorio que representan entornos independientes en una canalización. En un CI/CD proceso completo, los repositorios utilizan un flujo de trabajo basado en enlaces troncales, en el que una rama actúa como la única fuente de información para las implementaciones de ese repositorio. Para obtener más información, consulte [Enfoque basado en enlaces troncales](#). Otros indicadores incluyen los pasos de implementación manual distintos de las simples decisiones de aprobación o rechazo, el uso de revisiones y los lanzamientos programados.

¿Qué sucede si quiero utilizar un CI/CD proceso completo, pero aun así quiero programar el lanzamiento de determinadas funciones para momentos específicos?

Esto se suele hacer con indicadores de características. En este proceso, las implementaciones se siguen efectuando de forma continua, pero algunas características se ocultan mediante cierres condicionales en el código hasta que llegue el momento de lanzarlas.

¿Qué sucede si algunos pasos de mi proceso de implementación no se pueden automatizar?

Uno de los objetivos de una programación completa es CI/CD minimizar la necesidad de procesos manuales, pero no cabe duda de que hay posibles casos de uso en los que pueden ser necesarios los procesos manuales. De hecho, los procesos de solo lectura, como la consulta de los registros de las aplicaciones, suelen ejecutarse en entornos de producción con un riesgo mínimo. Sin embargo, se recomienda encarecidamente utilizar las acciones de escritura manual en la producción como el último recurso.

¿Qué sucede si mi personal técnico se siente más cómodo con los flujos de trabajo tradicionales que con un CI/CD proceso completo?

Es habitual que el personal técnico se resista a los cambios importantes, especialmente cuando algo que solía ser una práctica recomendada se sustituye por algo más nuevo. La tecnología avanza rápido y las mejoras se descubren constantemente. Si bien un cierto grado de escepticismo es una buena cualidad para el personal técnico, es igual de importante que esté abierto a los cambios. No actúe demasiado rápido con el personal escéptico, ya que debe administrar los cambios en el sistema antes de implementarlos. La clave es evitar que los escépticos permanezcan estáticos para siempre.

¿Qué ocurre si mis entornos están en varias cuentas? ¿Puedo seguir utilizando un CI/CD proceso completo?

Sí, de hecho, se recomienda utilizar una cuenta independiente para cada entorno. Para obtener más información sobre una canalización que activa etapas en distintas cuentas, consulta [Crear una canalización CodePipeline que utilice recursos de otra Cuenta de AWS](#).

Pasos a seguir a continuación

Utilice la sección [Prueba de fuego para canalizaciones de CI/CD](#) para evaluar los procesos de DevOps de la organización. Determine si sus procesos son de CI/CD completa. Si no lo son, decida si es necesario mejorarlos para aprovechar al máximo los beneficios de las implementaciones de CI/CD.

¿Cómo sabe que ha terminado? La respuesta es que muchas organizaciones en realidad nunca terminan. Se detienen en algún punto del camino, en un lugar adecuado para su caso de uso. Si bien el mejor escenario posible es una canalización de CI/CD completa, depende en gran medida de la situación de la organización y de las partes interesadas que tomaron la decisión. Las partes interesadas deben decidir qué etapa de la implementación de CI/CD funciona mejor para su caso de uso y cuál es la mejor manera de planificar el progreso hacia las siguientes fases.

Para más información sobre el diseño y la creación de canalizaciones de CI/CD, consulte [Recursos](#).

Recursos

AWS documentación y referencias

- [AWS Deployment Pipeline Reference Architecture](#)
- [¿Qué es la entrega continua?](#)
- [Practicar la integración continua y la entrega continua AWS](#) (AWS documento técnico)
- [Cómo configurar una CI/CD canalización AWS\(tutorialAWS práctico\)](#)
- [Cree una canalización Regiones de AWS que no sea compatible AWS CodePipeline\(guíaAWS prescriptiva\)](#)
- [Arquitectura de referencia e implementaciones de referencia de los canales](#) de despliegue (AWS entrada del blog)

Herramientas y servicios

- [¿ CI/CD La prueba de fuego](#)
- [AWS Cloud Development Kit \(AWS CDK\)](#)
- [AWS CloudFormation](#)
- [AWS CodePipeline](#)

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	—	25 de agosto de 2023

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.

antipatronos

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 CCoE, 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 alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a 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 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 Regions de AWS your account can use](#).

regresión

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

volver a alojar

Consulte [Las 7 R](#).

versión

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

reubicar

Consulte [Las 7 R](#).

redefinir la plataforma

Consulte [Las 7 R](#).

recomprar

Consulte [Las 7 R](#).

resiliencia

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

política basada en recursos

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

matriz responsable, confiable, consultada e informada (RACI)

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

control receptivo

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

retain

Consulte [Las 7 R](#).

retirar

Consulte [Las 7 R](#).

Generación aumentada de recuperación (RAG)

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

rotación

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

control de acceso por filas y columnas (RCAC)

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

RPO

Consulte [objetivo de punto de recuperación](#).

RTO

Consulte [objetivo de tiempo de recuperación](#).

manual de procedimientos

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

S

SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión Consola de administración de AWS o llamar a las operaciones de la AWS API sin tener que crear un usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte [Acerca de la federación basada en SAML 2.0](#) en la documentación de IAM.

SCADA

Consulte [control de supervisión y adquisición de datos](#).

SCP

Consulte [política de control de servicio](#).

secreta

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

seguridad desde el diseño

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

control de seguridad

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

refuerzo de la seguridad

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

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

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

automatización de la respuesta de seguridad

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

cifrado del servidor

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

política de control de servicio (SCP)

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

punto de enlace de servicio

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

acuerdo de nivel de servicio (SLA)

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

indicador de nivel de servicio (SLI)

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

objetivo de nivel de servicio (SLO)

Métrica objetivo que representa el estado de un servicio medido mediante un [indicador de nivel de servicio](#).

modelo de responsabilidad compartida

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

SIEM

Consulte [sistema de administración de eventos e información de seguridad](#).

único punto de error (SPOF)

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

SLA

Consulte [acuerdo de nivel de servicio](#).

SLI

Consulte [indicador de nivel de servicio](#).

SLO

Consulte [objetivo de nivel de servicio](#).

split-and-seed modelo

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para

crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para más información, consulte [Phased approach to modernizing applications in the Nube de AWS](#).

SPOF

Consulte [único punto de error](#).

esquema en estrella

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

patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda 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.

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.