Marco de ciclo de vida de resiliencia

AWS Guía prescriptiva

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Guía prescriptiva: Marco de ciclo de vida de resiliencia

Copyright © 2025 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
Términos y definiciones 2
Resiliencia continua 3
Etapa 1: Establecer objetivos 4
Mapeo de aplicaciones críticas 4
Mapeo de historias de usuarios 5
Definir las medidas 6
Creación de medidas adicionales 6
Etapa 2: Diseñar e implementar 8
AWS Marco Well-Architected 8
Comprender las dependencias 9
Estrategias de recuperación ante desastres 9
Definición de estrategias de CI/CD 10
¿Conducir ORRs 12
Comprender los límites del aislamiento de AWS fallas 12
Selección de respuestas 12
Modelado de resiliencia 13
Fallar de forma segura 14
Etapa 3: Evaluar y probar 15
Actividades previas al despliegue 15
Diseño del entorno 15
Prueba de integración 16
Canalizaciones de despliegue automatizadas 16
Prueba de carga 17
Actividades posteriores a la implementación 17
Realizar evaluaciones de resiliencia 17
pruebas de DR 18
Detección de desviaciones 18
Pruebas sintéticas 19
¿Ingeniería del caos? 19
Etapa 4: Operar 21
Observabilidad 21
Gestión de eventos 22
Resiliencia continua 22

Etapa 5: Responder y aprender ... 24
Creación de informes de análisis de incidentes ... 24
Realizar revisiones operativas ... 25
Revisar el rendimiento de las alarmas ... 26
Precisión de las alarmas ... 26
Falsos positivos ... 26
Falsos negativos ... 27
Alertas duplicadas ... 27
Realización de revisiones de métricas ... 27
Proporcionar formación y capacitación ... 27
Crear una base de conocimientos sobre incidentes ... 28
Implementar la resiliencia en profundidad ... 28
Conclusión y recursos ... 30
Colaboradores ... 31
Historial de documentos ... 32
Glosario ... 33
... 33
A ... 34
B ... 37
C ... 39
D ... 42
E ... 46
F ... 49
G ... 51
H ... 52
I ... 53
L ... 56
M ... 57
O ... 61
P ... 64
Q ... 67
R ... 67
S ... 70
T ... 74
U ... 76
V ... 77

W 77
Z 78
..... lxxx

Marco del ciclo de vida de la resiliencia: un enfoque continuo para la mejora de la resiliencia

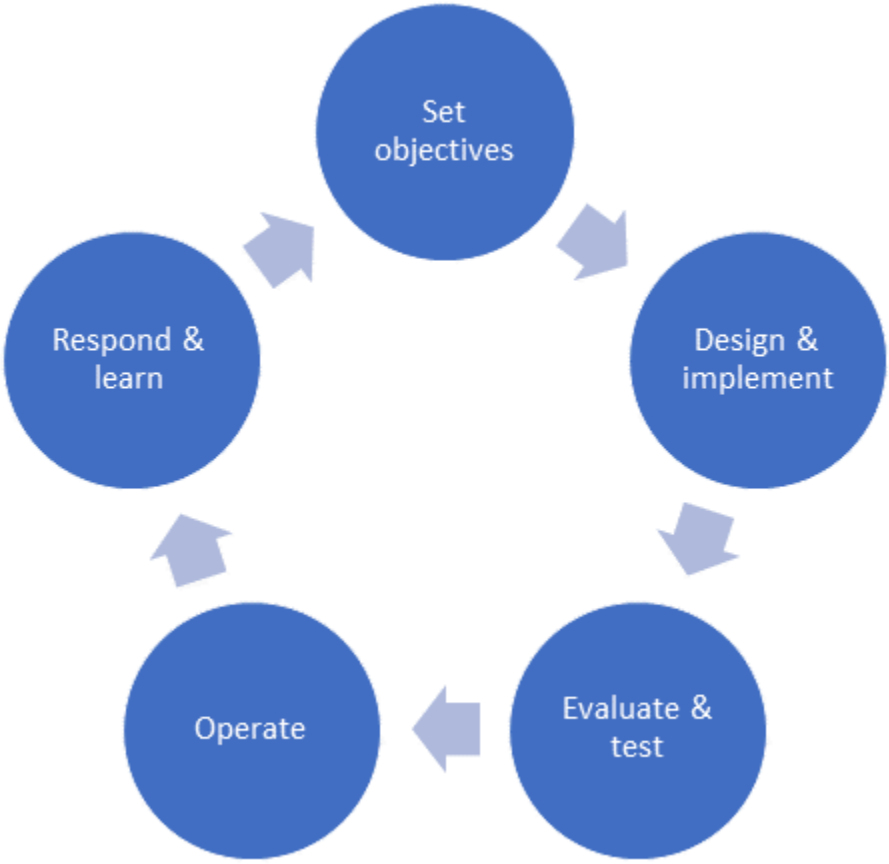
Amazon Web Services (colaboradores)

Octubre de 2023 (historial del documento)

Hoy en día, las organizaciones modernas se enfrentan a un número cada vez mayor de desafíos relacionados con la resiliencia, especialmente a medida que las expectativas de los clientes cambian hacia una mentalidad de estar siempre activos y disponibles. Los equipos remotos y las aplicaciones distribuidas y complejas se combinan con una creciente necesidad de versiones frecuentes. Como resultado, una organización y sus aplicaciones deben ser más resilientes que nunca.

AWS define la resiliencia como la capacidad de una aplicación para resistir las interrupciones o recuperarse de ellas, incluidas las relacionadas con la infraestructura, los servicios dependientes, las configuraciones erróneas y los problemas transitorios de la red. (Consulte la resiliencia y los componentes de la confiabilidad en la documentación del pilar de confiabilidad de AWS Well-Architected Framework). Sin embargo, para lograr el nivel de resiliencia deseado, a menudo es necesario hacer concesiones. La complejidad operativa, la complejidad de ingeniería y el costo deberán evaluarse y ajustarse en consecuencia.

Tras años de trabajo con clientes y equipos internos, AWS ha desarrollado un marco del ciclo de vida de la resiliencia que recoge los aprendizajes y las mejores prácticas en materia de resiliencia. El marco describe cinco etapas clave que se ilustran en el siguiente diagrama. En cada etapa, puede utilizar estrategias, servicios y mecanismos para mejorar su postura de resiliencia.



Estas etapas se analizan en las siguientes secciones de esta guía:

- Etapa 1: Establecer objetivos
- Etapa 2: Diseñar e implementar
- Etapa 3: Evaluar y probar
- Etapa 4: Operar
- Etapa 5: Responda y aprenda

Términos y definiciones

Los conceptos de resiliencia de cada etapa se aplican en diferentes niveles, desde componentes individuales hasta sistemas completos. La implementación de estos conceptos requiere una definición clara de varios términos:

- Un componente es un elemento que desempeña una función y consta de recursos de software y tecnología. Algunos ejemplos de componentes incluyen la configuración del código, la

infraestructura, como las redes, o incluso los servidores, los almacenes de datos y las dependencias externas, como los dispositivos de autenticación multifactor (MFA).

- Una aplicación es un conjunto de componentes que aportan valor empresarial, como una tienda web orientada al cliente o el proceso de back-end que mejora los modelos de aprendizaje automático. Una aplicación puede consistir en un subconjunto de componentes en una sola AWS cuenta o puede ser un conjunto de varios componentes que abarquen varias regiones. Cuentas de AWS
- Un sistema es un conjunto de aplicaciones, personas y procesos necesarios para administrar una función empresarial determinada. Abarca la aplicación necesaria para ejecutar una función; los procesos operativos, como la integración y la entrega continuas (CI/CD), la observabilidad, la gestión de la configuración, la respuesta a incidentes y la recuperación ante desastres; y los operadores que gestionan dichas tareas.
- Una interrupción es un suceso que impide que la aplicación desempeñe su función empresarial de forma adecuada.
- El deterioro es el efecto que una interrupción tiene en una aplicación si no se mitiga. Las aplicaciones pueden verse afectadas si sufren una serie de interrupciones.

Resiliencia continua

El ciclo de vida de la resiliencia es un proceso continuo. Incluso dentro de la misma organización, sus equipos de aplicaciones pueden funcionar con diferentes niveles de integridad en cada etapa, en función de los requisitos de la aplicación. Sin embargo, cuanto más completa sea cada etapa, mayor será el nivel de resiliencia que tendrá su aplicación.

Debe pensar en el ciclo de vida de la resiliencia como un proceso estándar que su organización puede poner en práctica. AWS ha modelado intencionadamente el ciclo de vida de la resiliencia para que sea similar al ciclo de vida del desarrollo de software (SDLC), con el objetivo de incorporar la planificación, las pruebas y el aprendizaje en todos los procesos operativos mientras desarrolla y opera sus aplicaciones. Como ocurre con muchos procesos de desarrollo ágiles, el ciclo de vida de la resiliencia se puede repetir con cada iteración del proceso de desarrollo. Le recomendamos que profundice progresivamente en las prácticas de cada etapa del ciclo de vida a lo largo del tiempo.

Etapa 1: Establecer objetivos

Comprender qué nivel de resiliencia se necesita y cómo se medirá es la base de la fase de objetivos establecidos. Es difícil mejorar algo si no tienes un objetivo y no puedes medirlo.

No todas las aplicaciones necesitan el mismo nivel de resiliencia. Cuando establezca los objetivos, tenga en cuenta el nivel necesario para realizar las inversiones y las compensaciones correctas. Una buena analogía para esto es un automóvil: tiene cuatro neumáticos pero solo lleva un neumático de repuesto. La probabilidad de que se pinchen varias llantas durante un viaje es baja, y tener piezas de repuesto adicionales podría reducir otras características, como el espacio de carga o la eficiencia de combustible, por lo que es una compensación razonable.

Una vez definidos los objetivos, se implementan controles de observabilidad en etapas posteriores (etapa 2: diseño e implementación y etapa 4: operación) para comprender si los objetivos se están cumpliendo.

Mapeo de aplicaciones críticas

La definición de los objetivos de resiliencia no debería ser exclusivamente una conversación técnica. En su lugar, comience con un enfoque orientado a los negocios para comprender lo que debe ofrecer la aplicación y las consecuencias de su deterioro. Esta comprensión de los objetivos empresariales se extiende luego a áreas como la arquitectura, la ingeniería y las operaciones. Cualquier objetivo de resiliencia que defina puede aplicarse a todas sus aplicaciones, pero la forma en que se miden los objetivos suele variar en función de la función de la aplicación. Es posible que esté ejecutando una aplicación fundamental para la empresa y, si esta aplicación no funciona correctamente, su organización podría perder importantes ingresos o sufrir daños en su reputación. Como alternativa, es posible que tenga otra aplicación que no sea tan crítica y que pueda tolerar algunos tiempos de inactividad sin que ello afecte negativamente a la capacidad de la organización para hacer negocios.

Como ejemplo, piense en una aplicación de gestión de pedidos para una empresa minorista. Si los componentes de la aplicación de gestión de pedidos están estropeados y no funcionan correctamente, no se realizarán nuevas ventas. Esta empresa minorista también tiene una cafetería para sus empleados ubicada en uno de sus edificios. La cafetería tiene un menú en línea al que los empleados pueden acceder en una página web estática. Si esta página web deja de estar disponible, algunos empleados podrían quejarse, pero no necesariamente causaría un daño financiero a la empresa. Según este ejemplo, es probable que la empresa opte por fijar objetivos de resiliencia más

ambiciosos para la aplicación de gestión de pedidos, pero no realizará una inversión significativa para garantizar la resiliencia de la aplicación web.

Identificar las aplicaciones más críticas, dónde aplicar el mayor esfuerzo y dónde hacer concesiones es tan importante como poder medir la resiliencia de una aplicación en la producción. Para comprender mejor el impacto de la discapacidad, puede realizar un análisis del impacto empresarial (BIA). Un BIA proporciona un enfoque estructurado y sistemático para identificar y priorizar las aplicaciones empresariales críticas, evaluar los posibles riesgos e impactos e identificar las dependencias de apoyo. La BIA ayuda a cuantificar el costo del tiempo de inactividad de las aplicaciones más importantes de su organización. Esta métrica ayuda a describir cuánto costará si una aplicación específica se ve afectada y no puede completar su función. En el ejemplo anterior, si la aplicación de gestión de pedidos está dañada, la empresa minorista podría perder ingresos significativos.

Mapeo de historias de usuarios

Durante el proceso de BIA, es posible que descubra que una aplicación es responsable de más de una función empresarial o que una función empresarial requiere varias aplicaciones. Siguiendo el ejemplo anterior de una empresa minorista, la función de gestión de pedidos puede requerir aplicaciones independientes para el pago, la promoción y la fijación de precios. Si una aplicación falla, el impacto podría recaer en la empresa y en los usuarios que interactúan con la empresa. Por ejemplo, es posible que la empresa no pueda añadir nuevos pedidos, ofrecer acceso a promociones y descuentos o actualizar el precio de sus productos. Estas diferentes funciones que requiere la función de gestión de pedidos pueden depender de varias aplicaciones. Estas funciones también pueden tener múltiples dependencias externas, lo que hace que el proceso de lograr una resiliencia centrada exclusivamente en los componentes sea demasiado complejo. Una mejor manera de gestionar este escenario es centrarse en las historias de los usuarios, que describen la experiencia que los usuarios esperan al interactuar con una aplicación o un conjunto de aplicaciones.

Centrarse en las historias de los usuarios le ayuda a comprender qué aspectos de la experiencia del cliente son los más importantes, de modo que puede crear mecanismos de protección contra amenazas específicas. En el ejemplo anterior, una historia de usuario podría ser el proceso de pago, que implica la aplicación de pago y depende de la aplicación de precios. Otra historia de usuario podría consistir en la visualización de promociones, lo que implica la solicitud de promoción. Tras mapear las aplicaciones más importantes y sus historias de usuario, puede empezar a definir las métricas que utilizará para medir la resiliencia de esas historias de usuario. Estas métricas se pueden aplicar a toda una cartera o a historias de usuarios individuales.

Definir las medidas

Los objetivos de punto de recuperación (RPOs), los objetivos de tiempo de recuperación (RTOs) y los objetivos de nivel de servicio (SLOs) son medidas estándar del sector que se utilizan para evaluar la resiliencia de un sistema determinado. El RPO se refiere a la cantidad de pérdida de datos que la empresa puede tolerar en caso de una falla, mientras que el RTO es una medida de la rapidez con la que una aplicación debe volver a estar disponible después de una interrupción. Estas dos métricas se miden en unidades de tiempo: segundos, minutos y horas. También puede medir la cantidad de tiempo durante el cual la aplicación funciona correctamente; es decir, realiza sus funciones tal y como está diseñada y es accesible para sus usuarios. Estos datos SLOs detallan el nivel de servicio esperado que recibirán los clientes y se miden mediante métricas como el porcentaje (%) de solicitudes que se atienden sin errores en un tiempo de respuesta inferior a un segundo (por ejemplo, el 99,99% de las solicitudes recibirán una respuesta cada mes). El RPO y el RTO están relacionados con las estrategias de recuperación ante desastres, ya que se supone que habrá interrupciones en el funcionamiento de las aplicaciones y los procesos de recuperación, que van desde la restauración de las copias de seguridad hasta la reorientación del tráfico de usuarios. SLOs se abordan mediante la implementación de controles de alta disponibilidad, que tienden a reducir el tiempo de inactividad de una aplicación.

Las métricas de SLO se utilizan habitualmente en la definición de los acuerdos de nivel de servicio (SLAs), que son contratos entre proveedores de servicios y usuarios finales. SLAs suelen incluir compromisos financieros y detallan las sanciones que debe pagar el proveedor si no se cumplen estos acuerdos. Sin embargo, un SLA no es una medida de tu postura de resiliencia, y aumentar un SLA no hace que tu aplicación sea más resiliente.

Puede empezar a establecer sus objetivos en función de SLOs RPOs, y. RTOs Una vez que haya definido sus objetivos de resiliencia y haya obtenido una comprensión clara de sus objetivos de RPO y RTO, podrá AWS Resilience Hubrealizar una evaluación de su arquitectura para descubrir posibles puntos débiles relacionados con la resiliencia. AWS Resilience Hub evalúa la arquitectura de una aplicación AWS comparándola con las mejores prácticas de Well-Architected Framework y comparte una guía de remediación en el contexto de lo que debe mejorarse específicamente para cumplir sus objetivos de RTO y RPO definidos.

Creación de medidas adicionales

El RPO, el RTO y el RTO SLOs son buenos indicadores de resiliencia, pero también puedes pensar en los objetivos desde una perspectiva empresarial y definirlos en torno a las funciones de tu

aplicación. Por ejemplo, tu objetivo podría ser: los pedidos realizados por minuto se mantendrán por encima del 98% si la latencia entre mi interfaz y mi servidor aumenta un 40%.O bien: las transmisiones iniciadas por segundo permanecerán dentro de una desviación estándar con respecto a la media, incluso si se pierde un componente específico. También puede crear objetivos para reducir el tiempo medio de recuperación (MTTR) en los tipos de errores conocidos; por ejemplo: los tiempos de recuperación se reducirán un x% si se produce alguno de estos problemas conocidos. La creación de objetivos que se ajusten a las necesidades de la empresa le ayuda a anticipar los tipos de fallos que su aplicación debería tolerar. También le ayuda a identificar enfoques para reducir la probabilidad de que su aplicación se vea afectada.

Si piensa en el objetivo de seguir funcionando si pierde el 5% de las instancias que alimentan su aplicación, podría decidir si la aplicación debe preescalarse o tener la capacidad de escalarse lo suficientemente rápido como para soportar el tráfico adicional que se genere durante ese evento. O bien, puede decidir aprovechar diferentes patrones arquitectónicos, tal y como se describe en la sección Etapa 2: diseño e implementación.

También debe implementar medidas de observabilidad para sus objetivos comerciales específicos. Por ejemplo, puedes hacer un seguimiento de la tasa media de pedidos, del precio medio de los pedidos, del número medio de suscripciones u otras métricas que puedan proporcionar información sobre el estado de la empresa en función del comportamiento de tu aplicación. Al implementar capacidades de observabilidad en su aplicación, puede crear alarmas y tomar medidas si estas métricas superan los límites definidos. La observabilidad se trata con más detalle en la sección Etapa 4: Operar.

Etapa 2: Diseñar e implementar

En la etapa anterior, estableces tus objetivos de resiliencia. Ahora, en la etapa de diseño e implementación, intenta anticipar los modos de falla e identificar las opciones de diseño, guiándose por los objetivos que estableció en la etapa anterior. También define estrategias para la gestión de cambios y desarrolla el código de software y la configuración de la infraestructura. En las siguientes secciones, se destacan las prácticas AWS recomendadas que debe tener en cuenta al tener en cuenta las desventajas, como el costo, la complejidad y los gastos operativos.

AWS Marco Well-Architected

Al diseñar su aplicación en función de los objetivos de resiliencia deseados, debe evaluar varios factores y hacer concesiones en función de la arquitectura más óptima. Para crear una aplicación altamente resiliente, debe tener en cuenta aspectos de diseño, creación e implementación, seguridad y operaciones. El AWS Well-Architected Framework proporciona un conjunto de mejores prácticas, principios de diseño y patrones arquitectónicos para ayudarlo a diseñar aplicaciones resilientes. AWS Los seis pilares del AWS Well-Architected Framework proporcionan las mejores prácticas para diseñar y operar sistemas resilientes, seguros, eficientes, rentables y sostenibles. El marco proporciona una forma de medir sus arquitecturas de manera coherente con las mejores prácticas e identificar las áreas de mejora.

Los siguientes son ejemplos de cómo AWS Well-Architected Framework puede ayudarle a diseñar e implementar aplicaciones que cumplan sus objetivos de resiliencia:

- El pilar de la confiabilidad: el pilar de la confiabilidad enfatiza la importancia de crear aplicaciones que puedan funcionar de manera correcta y consistente, incluso durante fallas o interrupciones. Por ejemplo, AWS Well-Architected Framework recomienda utilizar una arquitectura de microservicios para hacer que las aplicaciones sean más pequeñas y sencillas, de modo que pueda diferenciar las necesidades de disponibilidad de los distintos componentes de la aplicación. También puede encontrar descripciones detalladas de las mejores prácticas para crear aplicaciones mediante la limitación, el reintento con una reducción exponencial, los fallos rápidos (reducción de carga), la idempotencia, el trabajo constante, los disyuntores y la estabilidad estática.
- Revisión exhaustiva: El Marco de AWS Buena Arquitectura fomenta una revisión exhaustiva de su arquitectura comparándola con las mejores prácticas y los principios de diseño. Proporciona una forma de medir sus arquitecturas de forma coherente e identificar las áreas de mejora.

- Gestión de riesgos: el marco de AWS Well-Architected le ayuda a identificar y gestionar los riesgos que podrían afectar a la fiabilidad de su aplicación. Al abordar los posibles escenarios de fallo de forma proactiva, puede reducir su probabilidad o el deterioro resultante.
- Mejora continua: La resiliencia es un proceso continuo, y el AWS Well-Architected Framework hace hincapié en la mejora continua. Al revisar y perfeccionar periódicamente su arquitectura y sus procesos en función de las directrices del AWS Well-Architected Framework, puede asegurarse de que sus sistemas se mantengan resilientes ante los desafíos y requisitos en evolución.

Comprender las dependencias

Comprender las dependencias de un sistema es clave para la resiliencia. Las dependencias incluyen las conexiones entre los componentes de una aplicación y las conexiones a componentes externos a la aplicación, como los servicios compartidos propiedad de terceros APIs y de la empresa. Comprender estas conexiones le ayuda a aislar y gestionar las interrupciones, ya que una avería en un componente puede afectar a otros componentes. Este conocimiento ayuda a los ingenieros a evaluar el impacto de las deficiencias y a planificar en consecuencia, además de garantizar que los recursos se utilicen de forma eficaz. Comprender las dependencias le ayuda a crear estrategias alternativas y a coordinar los procesos de recuperación. También le ayuda a determinar los casos en los que puede reemplazar una dependencia física por una dependencia blanda, de modo que su aplicación pueda seguir desempeñando su función empresarial cuando se produzca una disminución de la dependencia. Las dependencias también influyen en las decisiones sobre el equilibrio de carga y el escalado de las aplicaciones. Comprender las dependencias es fundamental a la hora de realizar cambios en la aplicación, ya que puede ayudarle a determinar los posibles riesgos e impactos. Estos conocimientos le ayudan a crear aplicaciones estables y resilientes, lo que contribuye a la gestión de errores, la evaluación del impacto, la recuperación de averías, el equilibrio de carga, el escalado y la gestión de cambios. Puede realizar un seguimiento de las dependencias manualmente o utilizar herramientas y servicios, por ejemplo, AWS X-Raypara comprender las dependencias de sus aplicaciones distribuidas.

Estrategias de recuperación ante desastres

Una estrategia de recuperación ante desastres (DR) desempeña un papel fundamental en la creación y el funcionamiento de aplicaciones resilientes, principalmente al garantizar la continuidad empresarial. Garantiza que las operaciones comerciales cruciales puedan persistir con el menor deterioro posible, incluso durante eventos catastróficos, lo que minimiza el tiempo de inactividad y la posible pérdida de ingresos. Las estrategias de recuperación ante desastres son esenciales para la

protección de datos, ya que suelen incorporar copias de seguridad y replicación de datos periódicas en varias ubicaciones, lo que ayuda a proteger la valiosa información empresarial y a evitar su pérdida total en caso de desastre. Además, muchos sectores están regulados por políticas que exigen que las empresas cuenten con una estrategia de recuperación ante desastres para proteger los datos confidenciales y garantizar que los servicios permanezcan disponibles durante un desastre. Al garantizar un deterioro mínimo del servicio, una estrategia de recuperación ante desastres también refuerza la confianza y la satisfacción de los clientes. Una estrategia de recuperación ante desastres bien implementada y practicada con frecuencia reduce el tiempo de recuperación después de un desastre y ayuda a garantizar que las aplicaciones se vuelvan a poner en funcionamiento rápidamente. Además, los desastres pueden generar costes considerables, no solo por la pérdida de ingresos debido al tiempo de inactividad, sino también por los gastos que supone la restauración de aplicaciones y datos. Una estrategia de recuperación ante desastres bien diseñada ayuda a protegerse contra estas pérdidas financieras.

La estrategia que elija dependerá de las necesidades específicas de su aplicación, de su RTO y RPO y de su presupuesto. AWS Elastic Disaster Recoveryes un servicio de resiliencia diseñado específicamente que puede utilizar para ayudar a implementar su estrategia de recuperación ante desastres tanto para aplicaciones locales como basadas en la nube.

Para obtener más información, consulte el sitio web sobre la recuperación ante desastres de las cargas de trabajo AWS y los aspectos básicos de AWS varias regiones. AWS

Definición de estrategias de CI/CD

Una de las causas más comunes de las deficiencias en las aplicaciones son los cambios en el código u otros cambios que alteran la aplicación desde un estado de funcionamiento conocido anteriormente. Si no abordas la gestión de cambios con cuidado, esto puede provocar problemas frecuentes. La frecuencia de los cambios aumenta las oportunidades de impacto. Sin embargo, hacer cambios con menos frecuencia da como resultado conjuntos de cambios más grandes, que tienen muchas más probabilidades de provocar un deterioro debido a su alta complejidad. Las prácticas de integración y entrega continuas (CI/CD) están diseñadas para que los cambios sean pequeños y frecuentes (lo que se traduce en un aumento de la productividad) y, al mismo tiempo, someten cada cambio a un alto nivel de inspección mediante la automatización. Algunas de las estrategias fundamentales son:

- Automatización total: el concepto fundamental de la CI/CD es automatizar los procesos de creación e implementación en la medida de lo posible. Esto incluye la creación, las pruebas, el despliegue e incluso la supervisión. Las canalizaciones automatizadas ayudan a reducir la

posibilidad de errores humanos, garantizan la coherencia y hacen que el proceso sea más fiable y eficiente.

- Desarrollo impulsado por pruebas (TDD): escriba las pruebas antes de escribir el código de la aplicación. Esta práctica garantiza que todo el código tenga pruebas asociadas, lo que mejora la fiabilidad del código y la calidad de la inspección automatizada. Estas pruebas se ejecutan en el proceso de CI para validar los cambios.
- Compromisos e integraciones frecuentes: anime a los desarrolladores a confirmar el código con frecuencia y a realizar integraciones con frecuencia. Los cambios pequeños y frecuentes son más fáciles de probar y depurar, lo que reduce el riesgo de problemas importantes. La automatización reduce el coste de cada confirmación e implementación, lo que permite realizar integraciones frecuentes.
- Infraestructura inmutable: trate sus servidores y otros componentes de la infraestructura como entidades estáticas e inmutables. Sustituya la infraestructura en lugar de modificarla en la medida de lo posible y cree una nueva infraestructura mediante un código que se pruebe e implemente a lo largo de su proceso.
- Mecanismo de reversión: siempre tenga una forma fácil, confiable y probada con frecuencia de revertir los cambios si algo sale mal. Poder volver rápidamente al estado correcto conocido anteriormente es esencial para la seguridad del despliegue. Puede ser un simple botón para volver al estado anterior, o puede estar completamente automatizado e iniciarse mediante alarmas.
- Control de versiones: mantenga todo el código, la configuración e incluso la infraestructura de la aplicación como código en un repositorio controlado por versiones. Esta práctica ayuda a garantizar que pueda realizar un seguimiento fácil de los cambios y revertirlos si es necesario.
- Implementaciones tipo Canary e implementaciones azul/verde: implementar primero las nuevas versiones de la aplicación en un subconjunto de la infraestructura, o mantener dos entornos (azul/verde), le permite verificar el comportamiento de un cambio en la producción y revertirlo rápidamente si es necesario.

La CI/CD no se basa solo en las herramientas, sino también en la cultura. Crear una cultura que valore la automatización, las pruebas y el aprendizaje de los fracasos es tan importante como implementar las herramientas y los procesos correctos. Los retrocesos, si se hacen muy rápido y con un impacto mínimo, no deben considerarse un fracaso sino una experiencia de aprendizaje.

¿Conducir ORRs

Una revisión de la preparación operativa (ORR) ayuda a identificar las brechas operativas y procedimentales. En Amazon, creamos ORRs para resumir lo aprendido de décadas de operación de servicios de gran escala en preguntas seleccionadas con orientación sobre las mejores prácticas. Una ORR recoge las lecciones aprendidas anteriormente y requiere que los nuevos equipos se aseguren de tener en cuenta estas lecciones en sus aplicaciones. ORRs puede proporcionar una lista de los modos de falla o las causas de falla que se pueden incluir en la actividad de modelado de resiliencia que se describe en la sección de modelado de resiliencia a continuación. Para obtener más información, consulte Operational Readiness Reviews (ORRs) en el sitio web de AWS Well-Architected Framework.

Comprender los límites del aislamiento de AWS fallas

AWS proporciona múltiples límites de aislamiento de fallas para ayudarlo a alcanzar sus objetivos de resiliencia. Puede utilizar estos límites para aprovechar el alcance predecible de la contención de impactos que proporcionan. Debe familiarizarse con el diseño de AWS los servicios mediante el uso de estos límites, de modo que pueda tomar decisiones intencionadas sobre las dependencias que seleccione para su aplicación. Para saber cómo usar los límites en su aplicación, consulte Límites de aislamiento de AWS fallas en el sitio AWS web.

Selección de respuestas

Un sistema puede responder a una alarma de diversas formas. Algunas alarmas pueden requerir la respuesta del equipo de operaciones, mientras que otras pueden activar mecanismos de autorreparación dentro de la aplicación. Puede decidir conservar las respuestas que podrían automatizarse como operaciones manuales para controlar los costes de la automatización o gestionar las limitaciones de ingeniería. Es probable que el tipo de respuesta a una alarma se seleccione en función del costo de implementar la respuesta, la frecuencia prevista de la alarma, la precisión de la alarma y las posibles pérdidas de negocio si no se responde en absoluto a la alarma.

Por ejemplo, cuando un proceso de servidor se bloquea, el sistema operativo puede reiniciarlo, o puede aprovisionarse un nuevo servidor y finalizar el anterior, o puede indicarse a un operador que se conecte remotamente al servidor y lo reinicie. Estas respuestas tienen el mismo resultado (reiniciar el proceso del servidor de aplicaciones), pero tienen niveles variables de costos de implementación y mantenimiento.

> **Note**
>
> Puede seleccionar varias respuestas para adoptar un enfoque de resiliencia exhaustivo. Por ejemplo, en el escenario anterior, el equipo de aplicación podría optar por implementar las tres respuestas con un intervalo de tiempo entre cada una. Si el indicador de proceso fallido del servidor sigue en estado de alarma después de 30 segundos, el equipo puede suponer que el sistema operativo no ha podido reiniciar el servidor de aplicaciones. Por lo tanto, podrían crear un grupo de escalado automático para crear un nuevo servidor virtual y restaurar el proceso del servidor de aplicaciones. Si el indicador sigue en estado de alarma después de 300 segundos, es posible que se envíe una alerta al personal operativo para que se conecte al servidor original e intente restaurar el proceso.

La respuesta que elijan el equipo de aplicaciones y la empresa debe reflejar el interés de la empresa por compensar los gastos operativos mediante una inversión inicial en tiempo de ingeniería. Debe elegir una respuesta (un patrón de arquitectura, como la estabilidad estática, un patrón de software, como un disyuntor, o un procedimiento operativo), teniendo en cuenta cuidadosamente las limitaciones y el mantenimiento previsto de cada opción de respuesta. Es posible que existan algunas respuestas estándar para guiar a los equipos de aplicaciones, de modo que pueda utilizar las bibliotecas y los patrones que administra su función de arquitectura centralizada como base para esta consideración.

Modelado de resiliencia

Los modelos de resiliencia documentan la forma en que una aplicación responderá a las diferentes interrupciones anticipadas. Al anticipar las interrupciones, su equipo puede implementar procesos de observabilidad, controles automatizados y recuperación para mitigar o prevenir las deficiencias a pesar de las interrupciones. AWS ha creado una guía para desarrollar un modelo de resiliencia mediante el marco de análisis de la resiliencia. Este marco puede ayudarlo a anticipar las interrupciones y su impacto en su aplicación. Al anticipar las interrupciones, puede identificar las mitigaciones necesarias para crear una aplicación fiable y resiliente. Le recomendamos que utilice el marco de análisis de resiliencia para actualizar su modelo de resiliencia con cada iteración del ciclo de vida de la aplicación. El uso de este marco en cada iteración ayuda a reducir los incidentes al anticipar las interrupciones durante la fase de diseño y probar la aplicación antes y después del despliegue en producción. Desarrollar un modelo de resiliencia mediante este marco le ayuda a garantizar el cumplimiento de sus objetivos de resiliencia.

Fallar de forma segura

Si no puede evitar las interrupciones, fracase de forma segura. Considere la posibilidad de crear su aplicación con un modo de funcionamiento a prueba de fallos predeterminado, en el que no se incurra en pérdidas comerciales significativas. Un ejemplo de estado a prueba de fallos para una base de datos sería utilizar de forma predeterminada las operaciones de solo lectura, en las que los usuarios no pueden crear ni modificar ningún dato. En función de la confidencialidad de los datos, es posible que incluso desee que la aplicación se apague de forma predeterminada y que ni siquiera realice consultas de solo lectura. Tenga en cuenta cuál debe ser el estado a prueba de fallos de su aplicación y utilice este modo de funcionamiento de forma predeterminada en condiciones extremas.

Etapa 3: Evaluar y probar

Durante la fase de evaluación y prueba del ciclo de vida, la aplicación o los cambios en una aplicación existente se diseñaron, pero aún no se pusieron en producción. En esta etapa, se implementan actividades para probar las prácticas que se han realizado en etapas anteriores y se evalúan los resultados. Es posible que la aplicación aún esté en desarrollo activo o que el desarrollo principal esté completo y que se esté probando la aplicación antes de lanzarla a producción. Durante esta etapa, te centras en desarrollar y ejecutar pruebas que confirmen o refuten las expectativas de que la aplicación cumpla los objetivos de resiliencia definidos. Además, desarrolla y prueba los procedimientos operativos del sistema. Los procedimientos de despliegue que desarrolló en la fase 2: diseño e implementación se ponen en práctica y se evalúan los resultados. Si bien estas actividades de prueba y evaluación comienzan durante esta parte del ciclo de vida, no terminan aquí. Las pruebas y la evaluación continúan a medida que se pasa a la etapa 4: operación.

La fase de evaluación y prueba se divide en dos fases: actividades previas al despliegue y actividades posteriores al despliegue. Las actividades previas a la implementación consisten en tareas que deben completarse antes de implementar la aplicación en cualquier entorno, incluida la implementación de nuevas versiones del software y la implementación inicial en un entorno de pruebas. Las actividades posteriores a la implementación se llevan a cabo una vez que el software se ha implementado en un entorno de prueba o producción. En las siguientes secciones, se analizan estas fases con más detalle.

Actividades previas al despliegue

Diseño del entorno

El entorno en el que se prueban y evalúan las aplicaciones influyen en el grado de minuciosidad con que se puede realizar la prueba y en la confianza que se deposita en que esos resultados reflejan con precisión lo que ocurrirá en la fase de producción. Es posible que pueda realizar algunas pruebas de integración localmente en máquinas de desarrolladores mediante servicios como Amazon DynamoDB (consulte Configuración de DynamoDB local en la documentación de DynamoDB). Sin embargo, en algún momento tendrá que realizar las pruebas en un entorno que replique su entorno de producción para obtener la máxima confianza en los resultados. Este entorno conllevará costes, por lo que le recomendamos que adopte un enfoque gradual o por etapas para sus entornos, de forma que los entornos similares a los de producción aparezcan más adelante.

Prueba de integración

Las pruebas de integración son el proceso de comprobar que un componente bien definido de una aplicación desempeña sus funciones correctamente cuando funciona con dependencias externas. Esas dependencias externas pueden ser otros componentes desarrollados a medida, AWS servicios que utilice para su aplicación, dependencias de terceros y dependencias locales. Esta guía se centra en las pruebas de integración que demuestran la resiliencia de la aplicación. Se supone que ya existen pruebas unitarias y de integración que demuestran la precisión funcional del software.

Le recomendamos que diseñe pruebas de integración que comprueben específicamente los patrones de resiliencia que ha implementado, como los patrones de los disyuntores o la reducción de carga (consulte la etapa 2: Diseño e implementación). Las pruebas de integración orientadas a la resiliencia suelen implicar aplicar una carga específica a la aplicación o introducir interrupciones intencionadas en el entorno mediante el uso de capacidades como ().AWS Fault Injection ServiceAWS FIS
Lo ideal sería ejecutar todas las pruebas de integración como parte de su proceso de CI/CD y asegurarse de ejecutar las pruebas cada vez que se ejecute el código. Esto le ayuda a detectar y reaccionar rápidamente ante cualquier cambio en el código o la configuración que suponga una infracción de sus objetivos de resiliencia. Las aplicaciones distribuidas a gran escala son complejas, e incluso los cambios más pequeños pueden afectar significativamente a la resiliencia de partes aparentemente no relacionadas de la aplicación. Intenta ejecutar tus pruebas en cada confirmación. AWS proporciona un excelente conjunto de herramientas para operar su canalización de CI/CD y otras DevOps herramientas. Para obtener más información, consulte la Introducción a DevOps AWS on en el AWS sitio web.

Canalizaciones de despliegue automatizadas

El despliegue y las pruebas en sus entornos de preproducción son una tarea repetitiva y compleja que es mejor dejar en manos de la automatización. La automatización de este proceso libera recursos humanos y reduce la posibilidad de errores. El mecanismo para automatizar este proceso a menudo se denomina canalización. Cuando cree su canalización, le recomendamos que configure una serie de entornos de prueba que se acerquen cada vez más a su configuración de producción. Utiliza esta serie de entornos para probar repetidamente la aplicación. El primer entorno proporciona un conjunto de capacidades más limitado que el entorno de producción, pero implica un costo significativamente menor. Los entornos posteriores deberían añadir servicios y ampliarse para reflejar mejor el entorno de producción.

Comience por realizar pruebas en el primer entorno. Una vez que las implementaciones superen todas las pruebas del primer entorno de prueba, deje que la aplicación se ejecute con cierta cantidad

de carga durante un período de tiempo para comprobar si se produce algún problema con el tiempo. Confirme que ha configurado la observabilidad correctamente (consulte Precisión de las alarmas más adelante en esta guía) para poder detectar cualquier problema que pueda surgir. Cuando este período de observación se haya completado correctamente, despliegue la aplicación en el siguiente entorno de pruebas y repita el proceso, añadiendo pruebas adicionales o cargándolas según lo permita el entorno. Una vez que haya probado suficientemente la aplicación de este modo, puede utilizar los métodos de implementación que configuró previamente para implementar la aplicación en producción (consulte Definir estrategias de CI/CD, anteriormente en esta guía). El artículo Automatizar despliegues seguros y sin intervención en la Amazon Builders' Library es un recurso excelente que describe cómo Amazon automatiza el despliegue de código. La cantidad de entornos que preceden a la implementación de producción variará en función de la complejidad de la aplicación y de los tipos de dependencias que tenga.

Prueba de carga

A primera vista, las pruebas de carga se parecen a las pruebas de integración. Se prueba una función discreta de la aplicación y sus dependencias externas para comprobar que funciona según lo esperado. Por lo tanto, las pruebas de carga van más allá de las pruebas de integración y se centran en el funcionamiento de la aplicación con cargas bien definidas. Las pruebas de carga requieren la verificación de la funcionalidad correcta, por lo que deben realizarse después de una prueba de integración exitosa. Es importante entender qué tan bien responde la aplicación a las cargas esperadas y cómo se comporta cuando la carga supera las expectativas. Esto le ayuda a comprobar que ha implementado los mecanismos necesarios para garantizar que la aplicación siga siendo resistente ante una carga extrema. Para obtener una guía completa sobre las pruebas de carga AWS, consulte las pruebas de carga distribuidas AWS en la biblioteca de AWS soluciones.

Actividades posteriores a la implementación

La resiliencia es un proceso continuo y la evaluación de la resiliencia de la aplicación debe continuar una vez que la aplicación se haya implementado. Los resultados de las actividades posteriores a la implementación, como las evaluaciones de resiliencia continuas, pueden requerir que vuelva a evaluar y actualizar algunas de las actividades de resiliencia que realizó al principio del ciclo de vida de la resiliencia.

Realizar evaluaciones de resiliencia

La evaluación de la resiliencia no termina después de implementar la aplicación en producción. Incluso si tiene procesos de implementación automatizados y bien definidos, los cambios a veces

pueden producirse directamente en un entorno de producción. Además, es posible que haya factores que aún no haya tenido en cuenta en la verificación de la resiliencia previa a la implementación. AWS Resilience Hubproporciona un lugar central en el que puede evaluar si la arquitectura implementada cumple con las necesidades de RPO y RTO definidas. Puede utilizar este servicio para realizar evaluaciones bajo demanda de la resiliencia de su aplicación, automatizar las evaluaciones e incluso integrarlas en sus herramientas de CI/CD, tal y como se explica en la entrada del AWS blog Cómo evaluar continuamente la resiliencia de las aplicaciones con y. AWS Resilience Hub AWS CodePipeline La automatización de estas evaluaciones es una práctica recomendada, ya que ayuda a garantizar que se evalúa continuamente su postura de resiliencia en la producción.

pruebas de DR

En la fase 2: diseño e implementación, desarrolló estrategias de recuperación ante desastres (DR) como parte de su sistema. Durante la fase 4, debe probar sus procedimientos de recuperación ante desastres para asegurarse de que su equipo esté totalmente preparado para cualquier incidente y de que sus procedimientos funcionen según lo previsto. Debe probar todos sus procedimientos de recuperación ante desastres, incluidas las de conmutación por error y recuperación, de forma periódica y revisar los resultados de cada ejercicio para determinar si los procedimientos del sistema deben actualizarse y, de qué manera, para obtener los mejores resultados posibles. Cuando prepare inicialmente su prueba de DR, prográmela con suficiente antelación y asegúrese de que todo el equipo sepa qué esperar, cómo se medirán los resultados y qué mecanismo de retroalimentación se utilizará para actualizar los procedimientos en función de los resultados. Una vez que domines las pruebas de DR programadas, considera la posibilidad de realizarlas sin previo aviso. Los desastres reales no ocurren según un cronograma, por lo que debes estar preparado para poner en práctica tu plan en cualquier momento. Sin embargo, «sin previo aviso» no significa que no haya sido planificado. Las partes interesadas clave aún deben planificar el evento para garantizar que se cuente con una supervisión adecuada y que los clientes y las aplicaciones críticas no se vean afectados negativamente.

Detección de desviaciones

Pueden producirse cambios imprevistos en la configuración de las aplicaciones de producción incluso cuando se cuenta con procedimientos de automatización bien definidos. Para detectar cambios en la configuración de la aplicación, debe disponer de mecanismos para detectar las desviaciones, es decir, las desviaciones con respecto a una configuración de referencia. Para obtener información sobre cómo detectar desviaciones en las AWS CloudFormation pilas, consulte Detectar cambios de configuración no gestionados en las pilas y los recursos en la documentación.

AWS CloudFormation Para detectar desviaciones en el AWS entorno de su aplicación, consulte Detectar y resolver desviaciones AWS Control Tower en la documentación. AWS Control Tower

Pruebas sintéticas

Las pruebas sintéticas son el proceso de crear software configurable que se ejecute en producción, de forma programada, para probar las aplicaciones de APIs forma que se simule la experiencia del usuario final. Estas pruebas a veces se denominan canarios, en referencia al uso original del término en la minería del carbón. Las pruebas sintéticas suelen proporcionar alertas tempranas cuando una aplicación sufre una interrupción, incluso si la avería es parcial o intermitente, como suele ocurrir con los fallos grises.

¿Ingeniería del caos?

La ingeniería del caos es un proceso sistemático que implica someter deliberadamente una aplicación a eventos disruptivos de manera que se mitigue el riesgo, monitorear de cerca su respuesta e implementar las mejoras necesarias. Su objetivo es validar o cuestionar las suposiciones sobre la capacidad de la aplicación para gestionar dichas interrupciones. En lugar de dejar estos eventos al azar, la ingeniería del caos permite a los ingenieros organizar experimentos en un entorno controlado, normalmente durante los períodos de poco tráfico y con el apoyo de ingeniería disponible para mitigarlos eficazmente.

La ingeniería del caos comienza con la comprensión de las condiciones normales de operación, conocidas como estado estacionario, de la aplicación en cuestión. A partir de ahí, se formula una hipótesis que detalla el comportamiento exitoso de la aplicación en presencia de una interrupción. Usted lleva a cabo el experimento, que implica la introducción deliberada de interrupciones, que incluyen, entre otras, la latencia de la red, los fallos del servidor, los errores del disco duro y el deterioro de las dependencias externas. A continuación, analiza los resultados del experimento y mejora la resiliencia de la aplicación en función de lo aprendido. El experimento sirve como una herramienta valiosa para mejorar varios aspectos de la aplicación, incluido su rendimiento, y descubre problemas latentes que, de otro modo, podrían haber permanecido ocultos. Además, la ingeniería del caos ayuda a revelar las deficiencias en la observabilidad y las herramientas de alarma, y ayuda a refinarlas. También contribuye a reducir el tiempo de recuperación y a mejorar las habilidades operativas. La ingeniería del caos acelera la adopción de las mejores prácticas y fomenta una mentalidad de mejora continua. En última instancia, permite a los equipos desarrollar y perfeccionar sus habilidades operativas mediante la práctica y la repetición regulares.

AWS recomienda que comience sus esfuerzos de ingeniería del caos en un entorno que no sea de producción. Puede usar AWS Fault Injection Service (AWS FIS) para ejecutar experimentos de

ingeniería del caos con fallas de uso general o con fallas exclusivas. AWS Este servicio totalmente gestionado incluye alarmas de parada y controles de permisos completos para que pueda adoptar fácilmente la ingeniería del caos con seguridad y confianza.

Etapa 4: Operar

Una vez que haya completado la fase 3: evaluación y prueba, estará listo para implementar la aplicación en producción. En la etapa de operación, usted implementa su aplicación en producción y administra la experiencia de sus clientes. El diseño y la implementación de la aplicación determinan muchos de sus resultados de resiliencia, pero esta etapa se centra en las prácticas operativas que el sistema utiliza para mantener y mejorar la resiliencia. Crear una cultura de excelencia operativa ayuda a crear estándares y coherencia en estas prácticas.

Observabilidad

La parte más importante de entender la experiencia del cliente es mediante el monitoreo y las alarmas. Hay que instrumentar la aplicación para entender su estado y se necesitan diversas perspectivas, lo que significa que hay que medir tanto desde el lado del servidor como desde el lado del cliente, por lo general, en el caso de las Islas Canarias. Sus métricas deben incluir datos sobre las interacciones de la aplicación con sus dependencias y dimensiones que se ajusten a sus límites de aislamiento de fallas. También debe generar registros que proporcionen detalles adicionales sobre cada unidad de trabajo realizada por la aplicación. Podrías considerar la posibilidad de combinar métricas y registros mediante una solución como el formato de métricas CloudWatch integrado de Amazon. Es probable que descubra que siempre desea una mayor observabilidad, así que considere las compensaciones de costo, esfuerzo y complejidad necesarias para implementar el nivel de instrumentación deseado.

Los siguientes enlaces proporcionan las mejores prácticas para instrumentar su aplicación y crear alarmas:

- Supervisión de los servicios de producción en Amazon (presentación de AWS re:Invent 2020)
- Amazon Builders' Library: la excelencia operativa en Amazon (presentación de re:Invent 2021)AWS
- Mejores prácticas de observabilidad en Amazon (presentación de AWS re:Invent 2022)
- Instrumentación de sistemas distribuidos para una visibilidad operativa (artículo de Amazon Builders' Library)
- Creación de cuadros de mando para una visibilidad operativa (artículo de Amazon Builders' Library)

Gestión de eventos

Deberías contar con un proceso de gestión de eventos para gestionar las deficiencias cuando tus alarmas (o, lo que es peor, tus clientes) te avisen de que algo va mal. Este proceso debe incluir la contratación de un operador de guardia, la intensificación de los problemas y el establecimiento de guías para adoptar enfoques coherentes de solución de problemas que ayuden a eliminar los errores humanos. Sin embargo, las deficiencias no suelen producirse de forma aislada; una sola aplicación podría afectar a muchas otras aplicaciones que dependen de ella. Puede abordar los problemas rápidamente si comprende todas las aplicaciones que se ven afectadas y reúne a los operadores de varios equipos en una sola teleconferencia. Sin embargo, según el tamaño y la estructura de su organización, este proceso puede requerir un equipo de operaciones centralizado.

Además de configurar un proceso de gestión de eventos, debes revisar periódicamente tus métricas a través de los paneles. Las revisiones periódicas le ayudan a comprender la experiencia del cliente y las tendencias a largo plazo en el rendimiento de su aplicación. Esto le ayuda a identificar los problemas y los cuellos de botella antes de que tengan un impacto significativo en la producción. Revisar las métricas de forma coherente y estandarizada ofrece beneficios importantes, pero requiere la participación de todos los interesados y una inversión de tiempo.

Los siguientes enlaces proporcionan las mejores prácticas para crear paneles de control y revisar las métricas operativas:

- Creación de cuadros de mando para una visibilidad operativa (artículo de Amazon Builders' Library)
- El enfoque de Amazon para fracasar con éxito (presentación de AWS re:Invent 2019)

Resiliencia continua

Durante la fase 2 (diseño e implementación) y la fase 3 (evaluación y prueba), se iniciaron las actividades de revisión y prueba antes de implementar la aplicación en producción. Durante la fase de operación, debe continuar iterando esas actividades en producción. Debe revisar periódicamente la postura de resiliencia de su aplicación mediante las revisiones del Marco de Arquitectura AWS Bien Arquitectada, las Revisiones de Preparación Operacional ORRs () y el marco de análisis de resiliencia. Esto ayuda a garantizar que su aplicación no se desvíe de las bases de referencia y los estándares establecidos y le mantiene al día con directrices nuevas o actualizadas. Estas actividades de resiliencia continua le ayudan a descubrir interrupciones imprevistas anteriormente y a idear nuevas medidas de mitigación.

También puedes plantearte la posibilidad de realizar experimentos de ingeniería del caos durante los días de juego y después de haberlos realizado con éxito en entornos de preproducción. Los días de juego simulan eventos conocidos para los que has creado mecanismos de resiliencia para mitigarlos. Por ejemplo, un día de juego podría simular una avería en el servicio AWS regional e implementar una conmutación por error en varias regiones. Si bien la implementación de estas actividades puede requerir un esfuerzo considerable, ambas prácticas le ayudan a tener la confianza de que su sistema es resistente a los modos de falla para los que lo ha diseñado.

Al operar sus aplicaciones, detectar eventos operativos, revisar las métricas y probar su aplicación, encontrará numerosas oportunidades para responder y aprender.

Etapa 5: Responder y aprender

La forma en que la aplicación responde a los eventos disruptivos influye en su fiabilidad. Aprender de la experiencia y de la forma en que su aplicación ha respondido a las interrupciones en el pasado también es fundamental para mejorar su fiabilidad.

La etapa Responda y aprenda se centra en las prácticas que puede implementar para responder mejor a los eventos disruptivos en sus aplicaciones. También incluye prácticas que le ayudarán a extraer el máximo provecho de las experiencias de sus ingenieros y equipos de operaciones.

Creación de informes de análisis de incidentes

Cuando se produce un incidente, la primera acción es evitar que los clientes y la empresa sufran más daños lo antes posible. Una vez que la aplicación se haya recuperado, el siguiente paso es entender qué ha ocurrido e identificar las medidas para evitar que vuelva a ocurrir. Este análisis posterior al incidente suele plasmarse en un informe que documenta el conjunto de eventos que provocaron el deterioro de la aplicación y los efectos de la interrupción en la aplicación, los clientes y la empresa. Estos informes se convierten en valiosos recursos de aprendizaje y deberían compartirse ampliamente en toda la empresa.

> **Note**
>
> Es fundamental realizar un análisis de los incidentes sin culpar a nadie. Suponga que todos los operadores tomaron el mejor y más apropiado curso de acción teniendo en cuenta la información de que disponían. No utilice los nombres de los operadores o ingenieros en un informe. Citar el error humano como motivo de deterioro puede provocar que los miembros del equipo actúen con cautela para protegerse a sí mismos, lo que podría provocar la captura de información incorrecta o incompleta.

Un buen informe de análisis de incidentes, como el documentado en el proceso de corrección de errores (COE) de Amazon, sigue un formato estandarizado e intenta capturar, con el mayor detalle posible, las condiciones que provocaron el deterioro de la aplicación. El informe detalla una serie de eventos con fecha determinada y captura datos cuantitativos (a menudo métricas y capturas de pantalla de los paneles de control) que describen el estado mensurable de la aplicación a lo largo del tiempo. El informe debe reflejar los procesos de pensamiento de los operadores e ingenieros que tomaron medidas y la información que les llevó a sacar sus conclusiones. El informe también

debe detallar el rendimiento de los diferentes indicadores, por ejemplo, qué alarmas se emitieron, si esas alarmas reflejaban con precisión el estado de la aplicación, el intervalo de tiempo entre los eventos y las alarmas resultantes y el tiempo necesario para resolver el incidente. La cronología también incluye los manuales de ejecución o las automatizaciones que se iniciaron y cómo ayudaron a la aplicación a recuperar un estado útil. Estos elementos del cronograma ayudan a su equipo a comprender la eficacia de las respuestas automatizadas y de los operadores, incluida la rapidez con la que abordaron el problema y su eficacia a la hora de mitigar la interrupción.

Esta imagen detallada de un acontecimiento histórico es una poderosa herramienta educativa. Los equipos deben almacenar estos informes en un repositorio central que esté disponible para toda la empresa para que otros puedan revisar los eventos y aprender de ellos. Esto puede mejorar la intuición de sus equipos sobre lo que puede salir mal en la producción.

Un repositorio de informes detallados de incidentes también se convierte en una fuente de material de formación para los operadores. Los equipos pueden utilizar un informe de incidentes para inspirarse en un día de juego de mesa o en directo, en el que los equipos reciben información que reproduce la cronología recogida en el informe. Los operadores pueden analizar el escenario con información parcial de la cronología y describir las medidas que tomarían. A continuación, el moderador del día del partido podrá orientar sobre la respuesta de la aplicación en función de las acciones del operador. Esto desarrolla las habilidades de solución de problemas de los operadores, para que puedan anticipar y solucionar los problemas con mayor facilidad.

Un equipo centralizado responsable de la fiabilidad de las aplicaciones debe mantener estos informes en una biblioteca centralizada a la que pueda acceder toda la organización. Este equipo también debe ser responsable de mantener la plantilla del informe y de capacitar a los equipos sobre cómo completar el informe de análisis de incidentes. El equipo de confiabilidad debe revisar periódicamente los informes para detectar tendencias en toda la empresa que puedan abordarse mediante bibliotecas de software, patrones de arquitectura o cambios en los procesos del equipo.

Realizar revisiones operativas

Como se explica en la etapa 4: Operate, las revisiones operativas son una oportunidad para revisar las versiones recientes de funciones, los incidentes y las métricas operativas. La revisión operativa también es una oportunidad para compartir lo aprendido a partir de los lanzamientos de funciones y los incidentes con la comunidad de ingenieros de su organización en general. Durante la revisión operativa, los equipos analizan los despliegues de funciones que se han revertido, los incidentes que se han producido y la forma en que se han gestionado. Esto brinda a los ingenieros de toda la organización la oportunidad de aprender de las experiencias de otros y de hacer preguntas.

Dirija sus reseñas operativas a la comunidad de ingenieros de su empresa para que puedan obtener más información sobre las aplicaciones de TI que gestionan la empresa y los tipos de problemas a los que pueden enfrentarse. Llevarán consigo estos conocimientos a la hora de diseñar, implementar e implementar otras aplicaciones para la empresa.

Revisar el rendimiento de las alarmas

Las alarmas, tal como se explicó en la fase de operación, pueden provocar alertas en el panel de control, la creación de tickets, el envío de correos electrónicos o la creación de llamadas a los operadores. Una aplicación tendrá numerosas alarmas configuradas para monitorear varios aspectos de su funcionamiento. Con el tiempo, la precisión y la eficacia de estas alarmas deberán revisarse para aumentar la precisión de las alarmas, reducir los falsos positivos y consolidar las alertas duplicadas.

Precisión de las alarmas

Las alarmas deben ser lo más específicas posible para reducir el tiempo que se debe dedicar a interpretar o diagnosticar la interrupción específica que causó la alarma. Cuando se activa una alarma en respuesta a una avería de la aplicación, los operadores que reciben y responden a la alarma deben interpretar primero la información que transmite la alarma. La información puede consistir en un simple código de error que indica una línea de acción, como un procedimiento de recuperación, o puede incluir líneas de los registros de la aplicación que hay que revisar para entender por qué se ha emitido la alarma. A medida que su equipo aprenda a utilizar una aplicación de forma más eficaz, debería refinar estas alarmas para que sean lo más claras y concisas posible.

No se pueden anticipar todas las posibles interrupciones en una aplicación, por lo que siempre habrá alarmas generales que requieren que un operador las analice y diagnostique. Su equipo debería trabajar para reducir el número de alarmas generales a fin de mejorar los tiempos de respuesta y reducir el tiempo medio de reparación (MTTR). Lo ideal sería que hubiera una one-to-one relación entre una alarma y una respuesta automática o realizada por una persona.

Falsos positivos

Con el tiempo, los operadores ignorarán las alarmas que no requieran ninguna acción por parte de los operadores, pero que generen alertas en forma de correos electrónicos, páginas o tickets. Revise periódicamente las alarmas, o como parte de un análisis de incidentes, para identificar aquellas que suelen ignorarse o que no requieren ninguna acción por parte de los operadores (falsos

positivos). Debe esforzarse por eliminar la alarma o mejorarla para que emita una alerta procesable para los operadores.

Falsos negativos

Durante un incidente, las alarmas que están configuradas para alertar durante el incidente pueden fallar, tal vez debido a un evento que afecte a la aplicación de forma inesperada. Como parte del análisis de un incidente, debe revisar las alarmas que deberían haberse emitido pero que no se activaron. Deberías esforzarte por mejorar estas alarmas para que reflejen mejor las condiciones que podrían surgir a raíz de un evento. Como alternativa, es posible que tenga que crear alarmas adicionales que se refieran a la misma interrupción, pero que se activen por un síntoma diferente de la interrupción.

Alertas duplicadas

Es probable que una interrupción que afecte a la aplicación provoque varios síntomas y provoque varias alarmas. Periódicamente, o como parte de un análisis de incidentes, debe revisar las alarmas y alertas que se emitieron. Si los operadores recibieron alertas duplicadas, cree alarmas agregadas para consolidarlas en un único mensaje de alerta.

Realización de revisiones de métricas

Su equipo debe recopilar métricas operativas sobre su aplicación, como el número de incidentes por gravedad por mes, el tiempo que se tarda en detectar el incidente, el tiempo que se tarda en identificar la causa, el tiempo que se tarda en subsanar y el número de tickets creados, de alertas enviadas y de páginas generadas. Revise estas métricas al menos una vez al mes para comprender la carga que supone para el personal operativo, la signal-to-noise proporción a la que se enfrentan (por ejemplo, alertas informativas frente a alertas procesables) y si el equipo está mejorando su capacidad para operar las aplicaciones que están bajo su control. Utilice esta revisión para comprender las tendencias en los aspectos mensurables del equipo de operaciones. Solicita ideas al equipo sobre cómo mejorar estas métricas.

Proporcionar formación y capacitación

Es difícil capturar una descripción detallada de una aplicación y su entorno que provocó un incidente o un comportamiento inesperado. Además, modelar la resiliencia de la aplicación para anticipar estos escenarios no siempre es sencillo. Su organización debería invertir en materiales de capacitación y

capacitación para que sus equipos de operaciones y desarrolladores participen en actividades como la modelización de la resiliencia, el análisis de incidentes, las jornadas de juego y los experimentos de ingeniería del caos. Esto mejorará la fidelidad de los informes que producen sus equipos y los conocimientos que recopilan. Los equipos también estarán mejor preparados para anticipar los fallos sin tener que depender de un grupo de ingenieros más reducido y con más experiencia, que deberán aportar sus conocimientos mediante revisiones programadas.

Crear una base de conocimientos sobre incidentes

Un informe de incidentes es un resultado estándar de un análisis de incidentes. Debe utilizar el mismo informe o uno similar para documentar los escenarios en los que haya detectado un comportamiento anómalo de una aplicación, incluso si la aplicación no se ha deteriorado. Usa la misma estructura de informes estandarizada para recopilar el resultado de los caóticos experimentos y de los días de juego. El informe representa una instantánea de la aplicación y su entorno que provocó un incidente o un comportamiento inesperado. Debe almacenar estos informes estandarizados en un repositorio central al que puedan acceder todos los ingenieros de la empresa.

Luego, los equipos de operaciones y los desarrolladores pueden buscar en esta base de conocimientos para comprender qué ha interrumpido las aplicaciones en el pasado, qué tipos de situaciones podrían haber provocado la interrupción y qué evitó el deterioro de las aplicaciones. Esta base de conocimientos se convierte en un acelerador para mejorar las habilidades de sus equipos de operaciones y desarrolladores, y les permite compartir sus conocimientos y experiencias. Además, puedes utilizar los informes como material de formación o como escenarios para los días de juego o para hacer experimentos caóticos, a fin de mejorar la intuición del equipo operativo y su capacidad para solucionar problemas.

> **Note**
>
> Un formato de informe estandarizado también proporciona a los lectores una sensación de familiaridad y les ayuda a encontrar la información que buscan más rápidamente.

Implementar la resiliencia en profundidad

Como se mencionó anteriormente, una organización avanzada implementará múltiples respuestas a una alarma. No hay garantía de que una respuesta sea efectiva, por lo que, si se agrupan las respuestas por capas, una aplicación estará mejor preparada para fallar sin problemas. Te

recomendamos que implementes al menos dos respuestas para cada indicador a fin de garantizar que una respuesta individual no se convierta en un único punto de error que pueda desembocar en un escenario de recuperación ante desastres. Estas capas deben crearse en orden de serie, de modo que solo se ejecute una respuesta sucesiva si la respuesta anterior no fue efectiva. No debes ejecutar varias respuestas en capas a una sola alarma. En su lugar, utilice una alarma que indique si una respuesta no ha tenido éxito y, de ser así, inicie la siguiente respuesta en capas.

Conclusión y recursos

Esta guía presenta un ciclo de vida que le ayuda a mejorar continuamente la resiliencia de sus aplicaciones mediante la implementación de las mejores prácticas en cinco etapas: establecer objetivos, diseñar e implementar, evaluar y probar, operar y responder y aprender.

Para obtener más información sobre los servicios y conceptos que se analizan en esta guía, consulte los siguientes recursos.

AWS servicios:

- AWS Backup
- AWS Elastic Disaster Recovery
- AWS Fault Injection Service (AWS FIS)
- AWS Resilience Hub
- Controlador de recuperación de aplicaciones (ARC) de Amazon
- AWS X-Ray

Artículos y publicaciones de blog:

- La disponibilidad y más allá: comprender y mejorar la resiliencia de los sistemas distribuidos en AWS
- AWS Fault Isolation Boundaries
- AWS Aspectos fundamentales de varias regiones
- Ingeniería del caos en la nube
- Evaluar continuamente la resiliencia de las aplicaciones con AWS Resilience Hub y AWS CodePipeline
- Recuperación ante desastres de aplicaciones locales para AWS
- El pilar de la confiabilidad: AWS un marco de buena arquitectura
- Resilience analysis framework

Colaboradores

Entre los colaboradores de esta guía se encuentran:

- Bruno Emer, arquitecto principal de soluciones, AWS
- Clark Richey, arquitecto principal de soluciones, AWS
- Elaine Harvey, directora general de servicios de confiabilidad, AWS
- Jason Barto, arquitecto principal de soluciones, AWS
- John Formento, arquitecto principal de soluciones, AWS
- Lisi Lewis, directora sénior de marketing de productos, AWS
- Michael Haken, arquitecto principal de soluciones, AWS
- Neeraj Kumar, arquitecto principal de soluciones, AWS
- Wangechi Doble, arquitecto principal de soluciones, AWS

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	—	6 de octubre 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: migre su base de datos Oracle local a la edición compatible con PostgreSQL de Amazon Aurora.
- 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: migre su base de datos Oracle local a Amazon Relational Database Service (Amazon RDS) para Oracle en el. 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: migre su sistema de gestión de 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: migre su base de datos Oracle local a Oracle en una EC2 instancia del. 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 local 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 gestionados.

ACID

Consulte atomicidad, consistencia, aislamiento y 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 la 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 base de datos de origen gestiona las transacciones de las aplicaciones conectadas, mientras los datos se replican en la base de datos de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

función agregada

Función SQL que opera en un grupo de filas y calcula un único valor de retorno para el grupo. Algunos ejemplos de funciones agregadas incluyen `SUM` y`MAX`.

IA

Véase inteligencia artificial.

AIOps

Consulte las operaciones de inteligencia artificial.

anonimización

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

antipatrones

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

control de aplicaciones

Un enfoque de seguridad que permite el uso únicamente de 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

Un bot malo

Un bot destinado a interrumpir o causar daño a personas u organizaciones.

BCP

Consulte la 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. Véase también endianness.

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

Una estrategia de despliegue 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 en el otro entorno (verde). Esta estrategia le ayuda a revertirla rápidamente 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 información en Internet. Algunos otros bots, conocidos como bots malos, tienen como objetivo interrumpir o causar daños a personas u organizaciones.

botnet

Redes de bots que están infectadas por malware y que están bajo el control de una sola parte, conocida como pastor 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 con cristales rotos

En circunstancias excepcionales y mediante un proceso aprobado, un usuario puede acceder rápidamente a un sitio para el Cuenta de AWS que normalmente no tiene permisos de acceso. Para obtener más información, consulte el indicador Implemente procedimientos de rotura de cristales en la guía Well-Architected AWS .

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 el marco AWS de adopción de la nube.

despliegue canario

El lanzamiento lento e incremental de una versión para los usuarios finales. Cuando está seguro, despliega la nueva versión y reemplaza la versión actual en su totalidad.

CCoE

Consulte Cloud Center of Excellence.

CDC

Consulte la 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

Introducir intencionalmente 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 la integración continua y la entrega continua.

clasificación

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

cifrado del cliente

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

Centro de excelencia en la nube (CCoE)

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

computación en la nube

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

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 cuatro fases por las que suelen pasar las organizaciones cuando migran a Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realizar inversiones fundamentales para escalar su adopción de la nube (p. ej., crear una landing zone, definir una CCo E, establecer un modelo de operaciones)
- Migración: migración de aplicaciones individuales
- Reinvención: optimización de productos y servicios e innovación en la nube

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

CMDB

Consulte la base de datos de administración de la 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. Los repositorios en la nube más comunes incluyen GitHub oBitbucket 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 aprendizaje automático para analizar y extraer información de formatos visuales, como imágenes y vídeos digitales. Por ejemplo, Amazon SageMaker AI proporciona algoritmos de procesamiento de imágenes para CV.

desviación de configuración

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

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

Conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus comprobaciones 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 is commonly described as a pipeline. CI/CDpuede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar la calidad del código y entregar con mayor rapidez. 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

Vea la 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.

desviación 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 desviación de los 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

Un marco arquitectónico que proporciona una propiedad de datos distribuida y descentralizada con una administración y un gobierno centralizados.

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

Un sistema de administración de datos que respalde la inteligencia empresarial, como la analítica. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para consultas y 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 el 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 de 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 utilizan habitualmente para restringir consultas, filtrar y etiquetar 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)

La estrategia y el proceso que se utilizan para minimizar el tiempo de inactividad y la pérdida de datos ocasionados por 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 el 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 deriva

Seguimiento de las desviaciones con respecto a una configuración de referencia. 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 el mapeo del flujo de valor del desarrollo.

E

EDA

Consulte el análisis exploratorio de datos.

EDI

Véase 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 nube, la computación perimetral puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

intercambio electrónico de datos (EDI)

El intercambio automatizado de documentos comerciales entre organizaciones. Para obtener más información, consulte Qué es el intercambio electrónico de datos.

cifrado

Proceso informático que transforma datos de texto plano, 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 el punto final del servicio.

servicio de punto de conexión

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

Un sistema que automatiza y gestiona los procesos empresariales clave (como la contabilidad, el MES y la gestión 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 una canalización de CI/CD, 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 datos

La tabla central de un esquema en forma de estrella. Almacena datos cuantitativos sobre las operaciones comerciales. Normalmente, 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.

fallan rápidamente

Una filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de un enfoque ágil.

límite de aislamiento de fallas

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 obtener más información, consulte Límites de AWS aislamiento de errores.

rama de característica

Consulte la sucursal.

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.

indicaciones de unos pocos pasos

Proporcionar a un LLM un pequeño número de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que realice una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, en el que los modelos aprenden a partir de ejemplos (planos) integrados en las instrucciones. Las indicaciones con pocas tomas pueden ser eficaces para tareas que requieren un formato, un razonamiento o un conocimiento del dominio específicos. Consulte también el apartado de mensajes sin intervención.

FGAC

Consulte el 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 modificados 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 el modelo básico.

modelo de base (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 obtener más información, consulte Qué son los modelos básicos.

G

IA generativa

Un subconjunto de modelos de IA que se han entrenado con grandes cantidades de datos y que pueden utilizar un simple mensaje de texto para crear contenido y artefactos nuevos, como imágenes, vídeos, texto y audio. Para obtener más información, consulte Qué es la IA generativa.

bloqueo geográfico

Consulta las 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 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, y el flujo de trabajo basado en enlaces troncales es el enfoque moderno preferido.

imagen dorada

Instantánea de un sistema o software que se utiliza 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 GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

H

HA

Consulte la 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 retenidos

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de aprendizaje automático. Puede utilizar los datos de reserva para evaluar el rendimiento del modelo comparando 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, las revisiones suelen realizarse fuera del flujo de trabajo habitual de las versiones. DevOps

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

Vea la infraestructura como código.

políticas basadas en identidad

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

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

Un modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar, aplicar parches o modificar la infraestructura existente. Las infraestructuras inmutables son intrínsecamente más consistentes, fiables y predecibles que las infraestructuras mutables. Para obtener más información, consulte las prácticas recomendadas para implementar con una infraestructura inmutable en Well-Architected Framework AWS .

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

Un término que Klaus Schwab introdujo en 2016 para referirse a la modernización de los procesos de fabricación mediante avances en la conectividad, los datos en tiempo real, la automatización, el análisis y la inteligencia artificial/aprendizaje automático.

infraestructura

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

infraestructura como código (IaC)

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

Internet de las cosas industrial (T) IIo

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

VPC de inspección

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

Internet de las cosas (IoT)

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

interpretabilidad

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

IoT

Consulte Internet de las cosas.

biblioteca de información de TI (ITIL)

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

administración de servicios de TI (ITSM)

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

ITIL

Consulte la 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 grande (LLM)

Un modelo de IA de aprendizaje profundo que se entrena previamente con una gran cantidad de datos. Un LLM puede realizar múltiples 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

Ver 7 Rs.

sistema little-endian

Un sistema que almacena primero el byte menos significativo. Véase también endianness.

LLM

Véase un modelo de lenguaje amplio.

entornos inferiores

Véase 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

Ver sucursal.

malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware puede 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 gestionados

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

sistema de ejecución de fabricación (MES)

Un sistema de software para rastrear, monitorear, documentar y controlar los procesos de producción que convierten las materias primas en productos terminados en el taller.

MAP

Consulte Migration Acceleration Program.

mecanismo

Un proceso completo en el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para realizar ajustes. Un mecanismo es un ciclo que se refuerza y mejora a 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 el sistema de ejecución de la fabricación.

Transporte telemétrico de Message Queue Queue (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: realoje la migración a Amazon EC2 con AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

Una herramienta en línea que proporciona información para validar el modelo de negocio para migrar a. 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 asociados 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

El enfoque utilizado para migrar una carga de trabajo a. Nube de AWS Para obtener más información, consulte la entrada de las 7 R de este glosario y consulte Movilice a su organización para acelerar las migraciones a gran escala.

ML

Consulte el aprendizaje automático.

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 obtener más información, consulte Estrategia para modernizar las aplicaciones en el 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 obtener más información, consulte Evaluación de la preparación para la modernización de las aplicaciones en el 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.

MAPA

Consulte la evaluación de la cartera de migración.

MQTT

Consulte Message Queue Queue Telemetría y Transporte.

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

Un modelo que actualiza y modifica la infraestructura existente 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 el control de acceso de origen.

OAI

Consulte la identidad de acceso de origen.

OCM

Consulte gestión 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

Véase el acuerdo a 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.

Comunicaciones de proceso abierto: arquitectura unificada (OPC-UA)

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

Una lista de preguntas y las mejores prácticas asociadas que le ayudan a comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles fallos. Para obtener más información, consulte Operational Readiness Reviews (ORR) en AWS Well-Architected Framework.

tecnología operativa (OT)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En la industria manufacturera, 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 el AWS CloudTrail que se registran todos los eventos para todos Cuentas de AWS los miembros 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 la revisión de la preparación operativa.

OT

Consulte la 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 la 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 la gestión del ciclo de vida del producto.

policy

Un objeto que puede definir los permisos (consulte la política basada en la identidad), especifique las condiciones de acceso (consulte la política basada en los recursos) o defina los permisos máximos para todas las cuentas de una organización AWS Organizations (consulte la política de control de servicios).

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. Para obtener más información, consulte Habilitación de la persistencia de datos en los microservicios.

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

Una condición de consulta que devuelve `true` o`false`, por lo general, se encuentra en una cláusula. `WHERE`

pulsar un predicado

Técnica de optimización de consultas de bases de datos que filtra los datos de la consulta antes de transferirlos. Esto reduce la cantidad de datos que se deben recuperar y procesar de la base de datos relacional y 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

Un 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

Un control de seguridad diseñado para evitar el despliegue de recursos no conformes. Estos controles escanean los recursos antes de aprovisionarlos. Si el recurso no cumple con el control, significa que no está aprovisionado. Para obtener más información, consulte la guía de referencia de controles en la AWS Control Tower documentación y consulte Controles proactivos en Implementación de controles de seguridad en AWS.

gestión del ciclo de vida del producto (PLM)

La gestión 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 el rechazo y la retirada.

entorno de producción

Consulte el entorno.

controlador lógico programable (PLC)

En la fabricación, una computadora adaptable y altamente confiable que monitorea las máquinas y automatiza los procesos de fabricación.

encadenamiento rápido

Utilizar la salida de una solicitud de LLM como entrada para la siguiente solicitud para generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en subtareas 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)

Un patrón que permite las 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 puedan suscribir otros microservicios. El sistema puede añadir 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

Véase responsable, responsable, consultado, informado (RACI).

RAG

Consulte Retrieval Augmented Generation.

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

Véase responsable, responsable, consultado, informado (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

Ver 7 Rs.

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

Ver 7 Rs.

Región

Una colección 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 obtener más información, consulte Regiones de AWS Especificar qué cuenta puede usar.

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 7 Rs.

versión

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

trasladarse

Ver 7 Rs.

redefinir la plataforma

Ver 7 Rs.

recompra

Ver 7 Rs.

resiliencia

La capacidad de una aplicación para resistir las interrupciones o recuperarse de ellas. La alta disponibilidad y la recuperación ante desastres son consideraciones comunes a la hora de planificar la resiliencia en el. Nube de AWS Para obtener más información, consulte Nube de AWS Resiliencia.

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 7 Rs.

jubilarse

Ver 7 Rs.

Generación aumentada de recuperación (RAG)

Tecnología de inteligencia artificial generativa en la que un máster hace referencia a una fuente de datos autorizada que se encuentra fuera de sus fuentes de datos de formación antes de generar una respuesta. Por ejemplo, un modelo RAG podría realizar una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para obtener más información, consulte Qué es el RAG.

rotación

Proceso de actualizar periódicamente un secreto para dificultar el acceso de un atacante 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 el objetivo del punto de recuperación.

RTO

Consulte el 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 AWS

Management Console 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 el control de supervisión y la adquisición de datos.

SCP

Consulte la política de control de servicios.

secreta

Información confidencial o restringida, como una contraseña o credenciales de usuario, que almacene de forma cifrada. AWS Secrets Manager Se compone del valor secreto y sus metadatos. El valor secreto puede ser binario, una sola cadena o varias cadenas. Para obtener más información, consulta ¿Qué hay en un secreto de Secrets Manager? en la documentación de Secrets Manager.

seguridad desde el diseño

Un 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 principales de controles de seguridad: preventivos, de detección, con capacidad 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

Una acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o remediarlo. Estas automatizaciones sirven como controles de seguridad detectables o adaptables que le ayudan a implementar las mejores prácticas AWS de seguridad. Algunos ejemplos de acciones de respuesta automatizadas incluyen la modificación de un grupo de seguridad de VPC, la aplicación de parches a una EC2 instancia de Amazon o la rotación de credenciales.

cifrado del servidor

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

Una 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 que compartes con respecto a la seguridad y AWS 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 la información de seguridad y el sistema de gestión de eventos.

punto único de fallo (SPOF)

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

SLA

Consulte el acuerdo de nivel de servicio.

SLI

Consulte el indicador de nivel de servicio.

SLO

Consulte el 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 obtener más información, consulte Enfoque gradual para modernizar las aplicaciones en el. Nube de AWS

SPOF

Consulte el punto único de falla.

esquema en forma de estrella

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

patrón de higo estrangulador

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

subred

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

supervisión, control y adquisición de datos (SCADA)

En la industria manufacturera, un sistema que utiliza hardware y software para monitorear 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

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

indicador del sistema

Una técnica para proporcionar contexto, instrucciones o pautas a un LLM para dirigir su comportamiento. Las indicaciones 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 ayudarle a administrar, identificar, organizar, buscar y filtrar recursos. 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 VPCs redes con las locales. Para obtener más información, consulte Qué es una pasarela de tránsito en la AWS Transit Gateway documentación.

flujo de trabajo basado en enlaces troncales

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

acceso de confianza

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

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

ajuste

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

equipo de dos pizzas

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

U

incertidumbre

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

tareas indiferenciadas

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

entornos superiores

Ver 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 realiza un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para procesar tareas, 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.

GUSANO

Mira, escribe una vez, lee muchas.

WQF

Consulte el marco AWS de calificación de la carga de trabajo.

escribe una vez, lee muchas (WORM)

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

Z

ataque de día cero

Un ataque, normalmente de malware, que aprovecha 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.

aviso de tiro cero

Proporcionar a un LLM instrucciones para realizar una tarea, pero sin ejemplos (imágenes) que puedan ayudar a guiarla. El LLM debe utilizar sus conocimientos previamente entrenados para

realizar la tarea. La eficacia de las indicaciones cero depende de la complejidad de la tarea y de la calidad de las indicaciones. Consulte también las indicaciones de pocos pasos.

aplicación zombi

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

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