Marco de ciclo de vida de resiliencia

AWS Guía prescriptiva

AWS Guía prescriptiva: Marco de ciclo de vida de resiliencia

Table of Contents

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

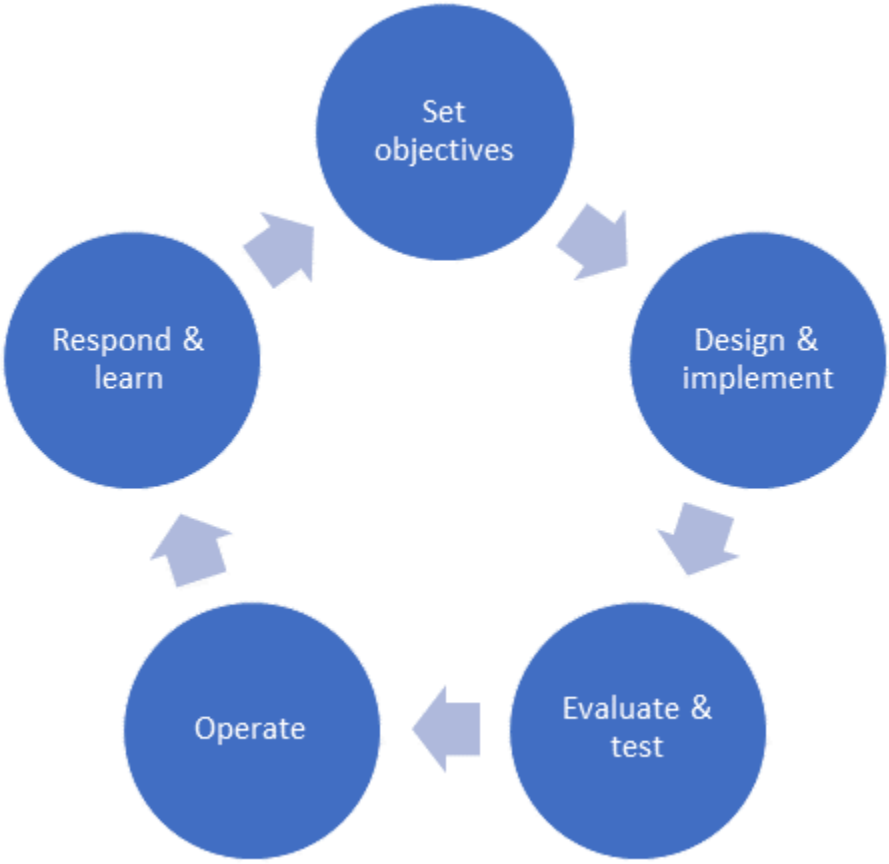
Amazon Web Services (colaboradores)

Octubre de 2023 (historia 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 activado y siempre disponible. Los equipos remotos y las aplicaciones distribuidas y complejas se combinan con una creciente necesidad de tener lanzamientos frecuentes. Como resultado, las organizaciones 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. Deberá evaluar y ajustar la complejidad operativa, la complejidad de ingeniería y el costo en consecuencia.

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



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

- Etapa 1: establecimiento de objetivos
- Etapa 2: diseño e implementación
- Etapa 3: evaluación y pruebas
- Etapa 4: funcionamiento
- Etapa 5: respuesta y aprendizaje

Términos y definiciones

Los conceptos de resiliencia de cada etapa se aplican en diferentes niveles, desde componentes individuales hasta sistemas completos. Para implementar estos conceptos es necesario hacer 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 son 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 ofrece valor comercial, como una tienda web orientada al cliente o el proceso de backend que mejora los modelos de machine learning. Una aplicación podría constar de un subconjunto de componentes en una sola cuenta de AWS o ser un conjunto de varios componentes que abarcan varias regiones y 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 administración de la configuración, la respuesta a incidentes y la recuperación ante desastres; y los operadores que administran dichas tareas.
- Una interrupción es un evento 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, los 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 la 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, a lo largo del tiempo, profundice progresivamente en las prácticas de cada etapa del ciclo de vida.

Etapa 1: establecimiento de objetivos

Entender el nivel de resiliencia que necesita y cómo se medirá es la base de la etapa de establecimiento de objetivos. Es difícil mejorar algo si no tiene un objetivo y no puede 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 concesiones correctas. Una buena analogía para este caso es un automóvil: tiene cuatro neumáticos pero solo lleva un neumático de repuesto. La probabilidad de que se pinchen varios neumáticos durante un viaje es baja, y tener más piezas de repuesto podría reducir otras características, como el espacio de carga o la eficiencia de combustible, por lo que es una concesión razonable.

Una vez definidos los objetivos, debe implementar controles de observabilidad en etapas posteriores (Etapa 2: diseño e implementación y Etapa 4: funcionamiento) para comprender si se cumplen los objetivos.

Asignación de aplicaciones críticas

La definición de los objetivos de resiliencia no debería ser una conversación exclusivamente técnica. En su lugar, comience con un enfoque orientado a la empresa para comprender qué 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. Puede aplicar cualquier objetivo de resiliencia que defina 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 ejecute una aplicación crítica 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 importante 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 administración de pedidos para una empresa minorista. Si los componentes de la aplicación de administración de pedidos están deteriorados y no funcionan correctamente, no se producirán 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 causaría un daño financiero a la empresa necesariamente. Según este ejemplo, es probable que la empresa opte por fijar objetivos más

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

Identificar las aplicaciones más importantes, dónde aplicar el mayor esfuerzo y dónde hacer concesiones es tan importante como poder medir la resiliencia de una aplicación en producción. Para comprender mejor el impacto de los deterioros, 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. El BIA ayuda a cuantificar el costo del tiempo de inactividad de las aplicaciones más importantes de la 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 administración de pedidos está deteriorada, la empresa minorista podría perder importantes ingresos.

Asignación de historias de usuarios

Durante el proceso del BIA, es posible que observe que una aplicación es responsable de más de una función empresarial o que una función empresarial necesita varias aplicaciones. Siguiendo el ejemplo anterior de una empresa minorista, la función de administración de pedidos puede requerir aplicaciones independientes para la tramitación de pedidos, las promociones y la fijación de precios. Si se produce un error en una aplicación, el impacto podría tener consecuencias para la empresa y los usuarios que interactúan con ella. Por ejemplo, es posible que la empresa no pueda agregar nuevos pedidos, ofrecer acceso a promociones y descuentos o actualizar el precio de los productos. Estas diferentes funciones que necesita la función de administración de pedidos pueden depender de varias aplicaciones. Estas funciones también pueden tener varias 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 esta situación es centrarse en las historias de usuarios, que describen la experiencia que los usuarios esperan al interactuar con una aplicación o un conjunto de aplicaciones.

Centrarse en las historias de usuarios lo 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 la tramitación del pedido, que implica la aplicación de tramitación de pedidos y depende de la aplicación de precios. Otra historia de usuario podría consistir en la visualización de promociones, lo que implica la aplicación de promociones. Tras asignar las aplicaciones más importantes y sus historias de usuarios, 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 específicas.

Definición de medidas

Los objetivos de punto de recuperación (RPO), los objetivos de tiempo de recuperación (RTO) y los objetivos de nivel de servicio (SLO) 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 un error, 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 SLO detallan el nivel de servicio esperado que recibirán los clientes y se miden con métricas como el porcentaje (%) de solicitudes que se atienden sin errores en un tiempo de respuesta inferior a un segundo (por ejemplo, cada mes recibirán una respuesta el 99,99 % de las solicitudes). El RPO y el RTO están relacionados con las estrategias de recuperación ante desastres, y suponen que se producirán interrupciones en el funcionamiento de las aplicaciones y los procesos de recuperación, que van desde la restauración de las copias de seguridad hasta la redirección del tráfico de usuarios. Los SLO se gestionan mediante la implementación de controles de alta disponibilidad, que tienden a reducir el tiempo de inactividad de una aplicación.

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

Puede empezar a establecer sus objetivos en función de los SLO, los RPO y los RTO. Una vez que haya definido sus objetivos de resiliencia y haya obtenido una comprensión clara de sus objetivos de RPO y RTO, puede usar AWS Resilience Hub para realizar una evaluación de su arquitectura para descubrir posibles puntos débiles relacionados con la resiliencia. AWS Resilience Hub evalúa la arquitectura de una aplicación comparándola con las prácticas recomendadas del Marco de AWS Well-Architected y comparte una guía de correcciones 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 los SLO son buenos indicadores de la resiliencia, pero también puede pensar en los objetivos desde una perspectiva empresarial y definirlos en torno a las funciones de la aplicación.

Por ejemplo, el objetivo podría ser: los pedidos realizados correctamente por minuto se mantendrán por encima del 98 % si la latencia entre mi frontend y mi backend aumenta un 40 %.O bien: los flujos iniciados 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 lo ayudará a anticipar los tipos de errores que la aplicación debe tolerar. También lo ayudará a identificar enfoques para reducir la probabilidad de que la aplicación se vea deteriorada.

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

También debe implementar medidas de observabilidad para los objetivos comerciales específicos. Por ejemplo, puede hacer un seguimiento de la tasa media de pedidos, el precio medio de los pedidos, el número medio de suscripciones u otras métricas que puedan proporcionar información sobre el estado de la empresa en función del comportamiento de la aplicación. Al implementar capacidades de observabilidad en la 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: funcionamiento.

Etapa 2: diseño e implementación

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

AWS Well-Architected Framework

Al diseñar la aplicación en función de los objetivos de resiliencia que desee, 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 Marco de AWS Well-Architected proporciona un conjunto de prácticas recomendadas, principios de diseño y patrones de arquitectura para ayudarlo a diseñar aplicaciones resilientes en AWS. Los seis pilares del Marco de AWS Well-Architected proporciona prácticas recomendadas para diseñar y utilizar sistemas resilientes, seguros, eficientes, rentables y sostenibles. El marco ofrece una forma de medir sus arquitecturas de forma constante en función de las prácticas recomendadas de arquitectura y de identificar áreas de mejora.

Los siguientes son ejemplos demuestran cómo el Marco de AWS Well-Architected puede ayudarlo a diseñar e implementar aplicaciones que cumplan sus objetivos de resiliencia:

- El pilar de la fiabilidad: el pilar de la fiabilidad enfatiza la importancia de crear aplicaciones que puedan funcionar de manera correcta y coherente, incluso durante errores o interrupciones. Por ejemplo, el Marco de AWS Well-Architected 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 prácticas recomendadas para crear aplicaciones mediante la limitación, el retroceso exponencial, la respuesta rápida a los errores (reducción de carga), la idempotencia, el trabajo constante, los disyuntores y la estabilidad estática.
- Revisión exhaustiva: el Marco de AWS Well-Architected fomenta una revisión exhaustiva de su arquitectura comparándola con las prácticas recomendadas y los principios de diseño. Proporciona una forma de medir sus arquitecturas de forma coherente y de identificar áreas de mejora.

- Administración de riesgos: el Marco de AWS Well-Architected lo ayuda a identificar y administrar los riesgos que podrían afectar a la fiabilidad de su aplicación. Al abordar las posibles situaciones de error de forma proactiva, puede reducir su probabilidad o el deterioro resultante.
- Mejora continua: la resiliencia es un proceso continuo y el Marco de AWS Well-Architected enfatiza la mejora continua. Al revisar y perfeccionar periódicamente su arquitectura y sus procesos en función de las directrices del Marco de AWS Well-Architected, puede asegurarse de que sus sistemas se mantengan resilientes ante los desafíos y requisitos en evolución.

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

Estrategias de recuperación ante desastres

Una estrategia de recuperación ante desastres (DR) desempeña un papel fundamental en la creación y la utilización de aplicaciones resilientes, principalmente al garantizar la continuidad empresarial. Garantiza que se puedan mantener las operaciones cruciales para la empresa con las mínimas averías posibles, incluso durante eventos catastróficos, lo que minimiza el tiempo de inactividad y la posible pérdida de ingresos. Las estrategias de recuperación ante desastres son

fundamentales para la protección de datos, ya que normalmente incorporan 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 la información confidencial 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 aplicada con frecuencia reduce el tiempo de recuperación después de un desastre y ayuda a garantizar que las aplicaciones vuelvan a estar disponibles rápidamente. Además, los desastres pueden generar costos considerables, no solo por la pérdida de ingresos debido al tiempo de inactividad, sino también por los gastos que supone la restauración de las aplicaciones y los 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, su RTO y RPO y su presupuesto. AWS Elastic Disaster Recovery es un servicio de resiliencia diseñado específicamente que puede utilizar para ayudar a implementar su estrategia de recuperación ante desastres tanto para las aplicaciones en las instalaciones como para las aplicaciones basadas en la nube.

Para obtener más información, consulte Disaster Recovery of Workloads on AWS y Aspectos fundamentales de las operaciones en múltiples regiones de AWS en el sitio web de AWS.

Definición de estrategias de CI/CD

Una de las causas más comunes de las averías en las aplicaciones son los cambios en el código u otros cambios que alteran la aplicación desde un estado de funcionamiento conocido anteriormente. No gestionar la administración de cambios con cuidado puede provocar averías frecuentes. La frecuencia de los cambios aumenta las oportunidades de que se produzca un impacto. Sin embargo, hacer cambios con menos frecuencia genera conjuntos de cambios de mayor tamaño, que tienen muchas más probabilidades de provocar una avería debido a su alta complejidad. Las prácticas de integración y entrega continuas (CI/CD) están diseñadas para hacer 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. Estas son algunas de las estrategias básicas:

- 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,

la implementación e incluso la supervisión. Los procesos automatizados ayudan a reducir la posibilidad de que se produzcan 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 la canalización de CI para validar los cambios.
- Confirmaciones e integraciones frecuentes: anime a los desarrolladores a confirmar el código y a realizar integraciones con frecuencia. Los cambios pequeños y frecuentes son más fáciles de probar y depurar, lo que reduce el riesgo de que surjan problemas importantes. La automatización reduce el costo de cada confirmación e implementación, lo que permite realizar integraciones frecuentes.
- Infraestructura inmutable: trate los servidores y demás componentes de la infraestructura como entidades estáticas e inmutables. En la medida de lo posible, sustituya la infraestructura en lugar de modificarla y cree la nueva infraestructura mediante código que se pruebe e implemente a lo largo de la canalización.
- Mecanismo de reversión: tenga siempre una forma fácil, fiable y probada con frecuencia de revertir los cambios si se produce algún problema. Poder volver rápidamente al estado correcto previo conocido es esencial para la seguridad de la implementación. Puede ser un simple botón para volver al estado anterior o puede automatizarse completamente 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 fácil seguimiento de los cambios y revertirlos si fuera necesario.
- Implementaciones canario 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 errores es tan importante como implementar las herramientas y los procesos correctos. Los retrocesos a versiones anteriores, si se hacen muy rápido y con un impacto mínimo, no deben considerarse un error sino una experiencia de aprendizaje.

Realización de ORR

Las revisiones de la preparación operativa (ORR) ayudan a identificar las brechas operativas y de procedimiento. En Amazon, creamos las ORR para resumir lo aprendido durante décadas de operación de servicios de gran escala en preguntas seleccionadas con orientación sobre las prácticas recomendadas. Una ORR recoge las lecciones aprendidas anteriormente y requiere que nuevos equipos se aseguren de haber implementado estas lecciones en sus aplicaciones. Las ORR pueden proporcionar una lista de los modos de error o las causas de error que se puede incluir en la actividad de modelado de resiliencia que se describe en la sección sobre los modelos de resiliencia que aparece a continuación. Para obtener más información, consulte Operational Readiness Reviews (ORRs) en el sitio web del Marco de AWS Well-Architected.

Descripción de los límites de aislamiento de errores de AWS

AWS proporciona varios límites de aislamiento de errores 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 los servicios de AWS mediante el uso de estos límites, de modo que pueda tomar decisiones intencionadas sobre las dependencias que elija para la aplicación. Para entender cómo usar los límites en la aplicación, consulte AWS Fault Isolation Boundaries en el sitio web de AWS.

Selección de respuestas

Un sistema puede responder a una alarma de distintas formas. Algunas alarmas pueden requerir una 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 costos de la automatización o administrar 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 para la empresa si no se responde en absoluto a la alarma.

Por ejemplo, cuando un proceso del servidor se bloquea, el sistema operativo puede reiniciarlo, o puede aprovisionarse un nuevo servidor y finalizar el anterior, o también puede indicarse a un operador que se conecte al servidor de forma remota y que lo reinicie. Estas respuestas tienen el mismo resultado (reiniciar el proceso del servidor de aplicaciones), pero varían en los 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 la aplicación podría optar por implementar las tres respuestas con un intervalo de tiempo entre cada una. Si el indicador de proceso del servidor con error sigue en estado de alarma después de 30 segundos, el equipo puede suponer que el sistema operativo no ha podido reiniciar el servidor de la aplicación. Por lo tanto, podrían crear un grupo de escalado automático para crear un nuevo servidor virtual y restaurar el proceso del servidor de la aplicación. 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 la aplicación 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.

Creación de modelos de resiliencia

En los modelos de resiliencia se documenta cómo responderá una aplicación 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 averías 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 la 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 la 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 de la implementación en producción. Desarrollar un modelo de resiliencia mediante este marco lo ayudará a garantizar el cumplimiento de sus objetivos de resiliencia.

Errores seguros

Si no puede evitar las interrupciones, haga que los errores sean seguros. Considere la posibilidad de crear la aplicación con un modo de funcionamiento a prueba de errores predeterminado, en el que no se incurra en pérdidas significativas para la empresa. Un ejemplo de estado a prueba de errores 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 se realicen consultas de solo lectura. Tenga en cuenta cuál debe ser el estado a prueba de errores de su aplicación y utilice este modo de funcionamiento de forma predeterminada en condiciones extremas.

Etapa 3: evaluación y pruebas

Durante la etapa de evaluación y pruebas del ciclo de vida, la aplicación o los cambios en una aplicación existente se han diseñado, pero aún no se han puesto 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 se haya completado y que se esté probando la aplicación antes de lanzarla a producción. Durante esta etapa, se centra en desarrollar y ejecutar pruebas que confirmen o refuten las expectativas de que la aplicación cumplirá los objetivos de resiliencia definidos. Además, desarrolla y prueba los procedimientos operativos del sistema. Pone en práctica los procedimientos de implementación que desarrolló en la etapa 2: diseño e implementación y evalúa 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: funcionamiento.

La etapa de evaluación y pruebas se divide en dos fases: actividades previas a la implementación y actividades posteriores a la implementación. Las actividades previas a la implementación comprenden las 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 cuando el software ya se ha implementado en un entorno de prueba o producción. En las siguientes secciones se tratan estas etapas de manera detallada.

Actividades previas a la implementación

Diseño del entorno

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

Prueba de integración

Las pruebas de integración son el proceso mediante el cual se comprueba que un componente bien definido de una aplicación desempeña sus funciones correctamente cuando funciona con dependencias externas. Esas dependencias externas podrían 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 hay pruebas unitarias y de integración que demuestran la precisión funcional del software.

Le recomendamos que diseñe pruebas de integración que prueben específicamente los patrones de resiliencia que ha implementado, como los patrones de los disyuntores o la reducción de carga (consulte Etapa 2: diseño e implementación). Habitualmente, en las pruebas de integración orientadas a la resiliencia se aplica una carga específica a la aplicación o se introducen interrupciones intencionadas en el entorno mediante el uso de capacidades como AWS Fault Injection Service (AWS FIS). Lo ideal es que ejecutes todas las pruebas de integración como parte de tu CI/CD proceso y te asegures de ejecutarlas cada vez que se confirma el código. Esto lo 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 de la aplicación que aparentemente no están relacionadas. Intenta ejecutar tus pruebas en cada confirmación. AWS proporciona un excelente conjunto de herramientas para operar su CI/CD canalización y otras DevOps herramientas. Para obtener más información, consulte la Introducción a DevOps AWS on en el AWS sitio web.

Canalizaciones de implementación automatizadas

La implementación 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 parezcan cada vez más a la configuración de producción. Debe utilizar esta serie de entornos para probar su aplicación de forma repetida. 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 agregar servicios y escalarse para reflejar mejor el entorno de producción.

Para empezar, realice pruebas en el primer entorno. Una vez que las implementaciones superen todas las pruebas del primer entorno de prueba, ejecute la aplicación con cierta cantidad de carga durante un periodo 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 periodo de observación se haya completado correctamente, implemente la aplicación en el siguiente entorno de pruebas y repita el proceso, agregando pruebas adicionales o cargándolas según lo permita el entorno. Una vez que haya probado suficientemente la aplicación de esta manera, puede utilizar los métodos de implementación que configuró previamente para implementar la aplicación en producción (consulte Definición de estrategias de CI/CD anteriormente en esta guía). El artículo Automating safe, hands-off deployments de la Amazon Builders' Library es un recurso excelente en el que se describe cómo Amazon automatiza la implementación 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. Pruebe una función discreta de la aplicación y sus dependencias externas para comprobar que funcione 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 del buen funcionamiento, por lo que deben realizarse después de una prueba de integración correcta. Es importante entender en qué medida responde bien la aplicación a las cargas esperadas y cómo se comporta cuando la carga supera las expectativas. Esto lo ayuda a comprobar que ha implementado los mecanismos necesarios para garantizar que la aplicación siga siendo resiliente 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 después de implementar la aplicación. 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 en materia de resiliencia que realizó al principio del ciclo de vida de la resiliencia.

Realización de evaluaciones de la resiliencia

La evaluación de la resiliencia no termina después de implementar la aplicación en producción. Incluso si tiene canalizaciones de implementación automatizadas y bien definidas, a veces pueden producirse cambios 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 Hub proporciona una ubicación central en la que puede evaluar si la arquitectura implementada cumple con las necesidades de RPO y RTO que ha definido. 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 AWS Resilience Hub y. 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 producción.

pruebas de DR

En la etapa 2: diseño e implementación, desarrolló estrategias de recuperación ante desastres (DR) como parte de su sistema. Durante la etapa 4, debe probar sus procedimientos de recuperación ante desastres para asegurarse de que el equipo esté totalmente preparado para cualquier incidente y de que los procedimientos funcionen según lo previsto. Debe probar todos sus procedimientos de recuperación ante desastres, incluidas la conmutación por error y la conmutación por recuperación, de forma periódica y revisar los resultados de cada ejercicio para determinar si debe actualizar los procedimientos del sistema y de qué manera para obtener los mejores resultados posibles. Cuando prepare inicialmente su prueba de recuperación ante desastres, prográmela con suficiente antelación y asegúrese de que todo el equipo sepa qué esperar, cómo se medirán los resultados y qué mecanismo para enviar comentarios se utilizará para actualizar los procedimientos en función de los resultados. Una vez que domine las pruebas de recuperación ante desastres programadas, considere la posibilidad de realizarlas sin previo aviso. Los desastres reales no se producen según una programación, por lo que debe estar preparado para poner en práctica su plan en cualquier momento. Sin embargo, “sin previo aviso” no significa que no se haya 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 esenciales 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 contar con mecanismos para detectar las derivas; es decir, las derivas con respecto a una configuración de referencia. Para obtener información sobre cómo detectar desviaciones en sus AWS CloudFormation pilas, consulte Detectar cambios de configuración no gestionados en pilas y 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 creación de 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 ser el caso de los errores 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, supervisar 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 estas interrupciones. En lugar de dejar estos eventos al azar, la ingeniería del caos permite que los ingenieros orquesten experimentos en un entorno controlado, normalmente durante los periodos de poco tráfico y con el apoyo de la ingeniería disponible, para mitigarlos de manera efectiva.

La ingeniería del caos comienza con la comprensión de las condiciones normales de funcionamiento (conocidas como estado estable) de la aplicación en cuestión. A partir de ahí, formula una hipótesis que detalla el comportamiento correcto de la aplicación cuando se produce una interrupción. Lleva a cabo el experimento, que implica la introducción deliberada de interrupciones, como, por ejemplo, la latencia de la red, los errores del servidor, los errores del disco duro y las averías en 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, como el rendimiento, y detecta 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 para las alarmas, 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 prácticas recomendadas y fomenta una mentalidad de mejora

continua. En última instancia, permite que los equipos desarrollen y perfeccionen 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 errores de uso general o con errores que sean exclusivos de AWS. Este servicio completamente administrado incluye alarmas con condiciones de detención y controles de permisos completos para que pueda adoptar fácilmente la ingeniería del caos con seguridad y confianza.

Etapa 4: funcionamiento

Una vez que haya completado la etapa 3: evaluación y pruebas, ya podrá implementar la aplicación en producción. En la etapa de funcionamiento, implementa la aplicación en producción y administra la experiencia de los 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. Establecer 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 la supervisión y las alarmas. Tiene que equipar la aplicación para entender su estado y necesita varias perspectivas, lo que significa que debe medir tanto desde el lado del servidor como del cliente, normalmente con canarios. Las métricas deben incluir datos sobre las interacciones de la aplicación con las dependencias y dimensiones que se ajusten a sus límites de aislamiento de errores. También debe generar registros que proporcionen más detalles sobre cada unidad de trabajo que realice la aplicación. Podría considerar la posibilidad de combinar métricas y registros mediante una solución como el formato de métricas integrado de Amazon CloudWatch. Es probable que observe que siempre necesita una mayor observabilidad, así que valore las concesiones necesarias en materia de costo, esfuerzo y complejidad para implementar el nivel de instrumentación deseado.

Los siguientes enlaces proporcionan las prácticas recomendadas para instrumentar la aplicación y crear alarmas:

- Monitoring production services at Amazon (presentación de AWS re:Invent 2020)
- Amazon Builders' Library: Operational Excellence at Amazon (presentación de AWS re:Invent 2021)
- Observability best practices at Amazon (presentación de AWS re:Invent 2022)
- Instrumentación de los sistemas distribuidos para obtener visibilidad operativa (artículo de Amazon Builders' Library)
- Building dashboards for operational visibility (artículo de Amazon Builders' Library)

Administración de eventos

Debe contar con un proceso de administración de eventos para gestionar las averías cuando las alarmas (o, lo que sería peor, los clientes) le indiquen que se ha producido algún problema. Este proceso debe incluir la contratación de un operador de guardia, la derivación de los problemas al equipo correspondiente y el establecimiento de manuales de procedimientos para adoptar enfoques coherentes de solución de problemas que ayuden a eliminar los errores humanos. Sin embargo, las averías no suelen producirse de forma aislada: una sola aplicación podría afectar a muchas otras aplicaciones que dependen de ella. Para abordar los problemas rápidamente, debe comprender 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 la organización, este proceso puede requerir un equipo de operaciones centralizado.

Además de configurar un proceso de administración de eventos, debe revisar periódicamente las métricas a través de paneles. Las revisiones periódicas lo ayudan a entender la experiencia del cliente y las tendencias a largo plazo en el rendimiento de la aplicación. Le servirán para identificar los problemas y los cuellos de botella antes de que tengan un impacto significativo en producción. Revisar las métricas de forma coherente y estandarizada ofrece ventajas importantes, pero requiere la participación de todas las partes interesadas y una inversión de tiempo.

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

- Building dashboards for operational visibility (artículo de Amazon Builders' Library)
- Amazon's approach to failing successfully (presentación de AWS re:Invent 2019)

Resiliencia continua

Durante la etapa 2: diseño e implementación y la etapa 3: evaluación y pruebas, inició las actividades de revisión y pruebas antes de implementar la aplicación en producción. Durante la etapa de funcionamiento, debe continuar iterando esas actividades en producción. Debe revisar periódicamente la postura de resiliencia de la aplicación mediante revisiones del Marco de AWS Well-Architected, revisiones de la preparación operativa (ORR) y el marco de análisis de la resiliencia. Esto ayuda a garantizar que la aplicación no se desvíe de las líneas de base y los estándares establecidos y le mantiene al día con directrices nuevas o actualizadas. Estas actividades de resiliencia continua lo ayudan a detectar interrupciones imprevistas y a idear nuevas medidas de mitigación.

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

Al poner en funcionamiento sus aplicaciones, detectar eventos operativos, revisar las métricas y probar la aplicación, encontrará numerosas oportunidades para responder y aprender.

Etapa 5: respuesta y aprendizaje

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

La etapa de respuesta y aprendizaje se centra en las prácticas que puede implementar para responder mejor a los eventos disruptivos que se producen en las aplicaciones. También incluye prácticas que lo ayudarán a obtener 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 consiste en evitar lo antes posible que los clientes y la empresa sufran más daños. Una vez que la aplicación se haya recuperado, el siguiente paso es entender lo que ha sucedido e identificar los pasos necesarios para evitar que vuelva a ocurrir. Este análisis posterior al incidente suele plasmarse en un informe que documenta el conjunto de eventos que provocaron la avería de la aplicación y los efectos de las interrupciones en la aplicación, los clientes y la empresa. Estos informes se convierten en valiosos artefactos de aprendizaje y deben compartirse ampliamente con toda la empresa.

> **Note**
>
> Es fundamental que realice un análisis de los incidentes sin culpar a nadie. Debe suponer que todos los operadores tomaron el mejor curso de acción y el más adecuado teniendo en cuenta la información de que disponían. No utilice los nombres de los operadores o ingenieros en los informes. Citar un error humano como motivo de las averías puede provocar que los miembros del equipo actúen con cautela para protegerse a sí mismos, lo que podría provocar que la información recopilada sea incorrecta o incompleta.

Los buenos informe de análisis de incidentes, como el que se documenta en el proceso de corrección de errores (COE) de Amazon, siguen un formato estandarizado e intentan capturar, con el mayor detalle posible, las condiciones que provocaron la avería en la aplicación. El informe detalla una serie de eventos con marca de tiempo y captura datos cuantitativos (a menudo, métricas y capturas de pantalla de los paneles de control de supervisión) que describen el estado medible

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 los llevó a sacar sus conclusiones. En el informe también se debe detallar el rendimiento de los diferentes indicadores; por ejemplo, qué alarmas se emitieron, si esas alarmas reflejaron 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 escala de tiempo también refleja los manuales de procedimientos o las automatizaciones que se iniciaron y cómo ayudaron a la aplicación a recuperar un estado útil. Estos elementos de la escala de tiempo ayudan a su equipo a comprender la eficacia de las respuestas automatizadas y de los operadores, como la rapidez con la que abordaron el problema y su eficacia a la hora de mitigar la interrupción.

Esta imagen detallada de un evento 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 otras personas 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.

Los repositorios de informes detallados de incidentes también se convierten en una fuente de material de formación para los operadores. Los equipos pueden utilizar los informes de incidentes como inspiración para un día de ejercicios prácticos o de juego en vivo en el que los equipos reciben información que reproduce la escala de tiempo recogida en el informe. Los operadores pueden repetir el escenario con información parcial de la escala de tiempo y describir las medidas que tomarían. A continuación, el moderador del día de juego podrá aportar orientación 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 anticiparse y solucionar los problemas con mayor facilidad.

Un equipo centralizado encargado 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 formar a los equipos sobre cómo rellenar el informe de análisis de incidentes. El equipo de fiabilidad tiene que revisar los informes periódicamente para detectar tendencias en toda la empresa que puedan abordarse mediante bibliotecas de software, patrones de arquitectura o cambios en los procesos del equipo.

Realización de revisiones operativas

Como se explica en la etapa 4: funcionamiento, las revisiones operativas son una oportunidad para revisar los recientes lanzamientos de características, los incidentes y las métricas operativas. La revisión operativa también es una oportunidad para compartir lo que se aprendió a partir de

los lanzamientos de características y los incidentes con toda la comunidad de ingenieros de su organización. Durante la revisión operativa, los equipos analizan las implementaciones de características que se han revertido, los incidentes que se han producido y la forma en que se han gestionado. De esta forma, los ingenieros de toda la organización tienen la oportunidad de aprender de las experiencias de otros y de formular preguntas.

Dirija sus revisiones operativas a la comunidad de ingenieros de la empresa de forma que puedan obtener más información sobre las aplicaciones de TI que hacen funcionar a la empresa y los tipos de problemas a los que pueden enfrentarse. Usarán estos conocimientos a la hora de diseñar, implementar y desplegar otras aplicaciones para la empresa.

Revisión el rendimiento de las alarmas

Las alarmas, tal como se explicó en la fase de funcionamiento, pueden generar alertas en el panel de control, la creación de tickets, el envío de correos electrónicos o la creación de llamadas a operadores. Una aplicación tendrá numerosas alarmas configuradas para supervisar 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 tendrá que dedicar a interpretar o diagnosticar la interrupción en concreto 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 primero tienen que interpretar la información que transmite la alarma. La información puede ser 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 activado 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 necesitarán que un operador las analice y diagnostique. Su equipo debería esforzarse por reducir la cantidad de alarmas generales a fin de mejorar los tiempos de respuesta y reducir el tiempo medio de reparación (MTTR). Lo ideal es que haya una relación unívoca entre una alarma y una respuesta automática o una respuesta a cargo de una persona.

Falsos positivos

Con el tiempo, los operadores ignorarán las alarmas que no requieran ninguna acción por su parte, pero que generen alertas en forma de correos electrónicos, páginas o tickets. Revise las alarmas periódicamente o como parte de un análisis de los incidentes para identificar aquellas que suelen ignorarse o que no requieren ninguna acción por parte de los operadores (falsos positivos). Debería esforzarse por eliminar la alarma o mejorarla para que emita una alerta útil para los operadores.

Falsos negativos

Durante un incidente, las alarmas que se han configurado para alertar durante el incidente podrían 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 activado pero que no se han emitido. Debería esforzarse 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 más alarmas que hagan referencia a la misma interrupción, pero que se activen debido a un síntoma diferente de la interrupción.

Alertas duplicadas

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

Realización de revisiones de métricas

El equipo debe recopilar métricas operativas sobre la aplicación, como el número de incidentes por gravedad al 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 corregir y el número de tickets creados, alertas enviadas y páginas generadas. Revise estas métricas al menos una vez al mes para comprender la carga que supone para el personal operativo, la relación señal/ruido a la que se enfrentan (por ejemplo, la relación entre alertas informativas y alertas útiles) y si el equipo está mejorando su capacidad para poner en funcionamiento las aplicaciones que están bajo su control. Utilice esta revisión para comprender las tendencias en los aspectos medibles del equipo de operaciones. Pida ideas al equipo sobre cómo mejorar estas métricas.

Prestación de formación y capacitación

Es difícil crear una descripción detallada de una aplicación y el entorno que provocó un incidente o un comportamiento inesperado. Además, crear modelos de resiliencia de la aplicación para anticipar estos escenarios no siempre es sencillo. Su organización debe invertir en materiales de formación y capacitación para que los equipos de operaciones y desarrolladores participen en actividades como la creación de modelos de 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 anticiparse a los errores sin tener que depender de un grupo de ingenieros más reducido y con más experiencia, que deberán aportar sus conocimientos mediante revisiones programadas.

Creación de 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 ha tenido ninguna avería. Use la misma estructura de informes estandarizada para recopilar el resultado de los experimentos del caos y de los días de juego. El informe representa una instantánea de la aplicación y el 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.

A continuación, 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 averías en la aplicación. 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, puede utilizar los informes como material de formación o como escenarios para los días de juego o para los experimentos del caos a fin de mejorar la intuición del equipo operativo y su capacidad para resolver las interrupciones.

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

Implementación de la resiliencia en profundidad

Como se mencionó anteriormente, las organizaciones avanzadas implementarán varias respuestas a una alarma. No hay ninguna garantía de que una respuesta sea efectiva, por lo que, si se agrupan las respuestas por capas, la aplicación estará mejor preparada para que se produzca un error sin causar problemas. Le recomendamos que implemente al menos dos respuestas para cada indicador a fin de garantizar que una respuesta específica 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 surtió efecto. No ejecute varias respuestas en capas ante una sola alarma. En su lugar, utilice una alarma que indique si una respuesta no ha funcionado correctamente y, de ser así, inicie la siguiente respuesta en capas.

Conclusión y recursos

En esta guía se presenta un ciclo de vida que lo ayuda a mejorar continuamente la resiliencia de sus aplicaciones al implementar las prácticas recomendadas en cinco etapas: establecimiento de objetivos, diseño e implementación, evaluación y pruebas, funcionamiento y respuesta y aprendizaje.

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

Servicios de AWS:

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

Artículos y entradas en el blog:

- Availability and Beyond: Understanding and Improving the Resilience of Distributed Systems on AWS
- AWS Fault Isolation Boundaries
- Aspectos fundamentales de las operaciones en múltiples regiones de AWS
- Chaos Engineering in the cloud
- Continually assessing application resilience with AWS Resilience Hub and AWS CodePipeline
- Disaster Recovery of On-Premises Applications to AWS
- Pilar de fiabilidad: Marco de AWS Well-Architected
- Resilience analysis framework

Colaboradores

Los colaboradores de esta guía son las siguientes personas:

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

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

A

ABAC

Consulte control de acceso basado en atributos.

servicios abstractos

Consulte servicios administrados.

ACID

Consulte atomicidad, consistencia, aislamiento, durabilidad.

migración activa-activa

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

migración activa-pasiva

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

función de agregación

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

IA

Consulte inteligencia artificial.

AIOps

Consulte operaciones de inteligencia artificial

anonimización

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

antipatrones

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

control de aplicaciones

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

cartera de aplicaciones

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

inteligencia artificial (IA)

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

operaciones de inteligencia artificial (AIOps)

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

cifrado asimétrico

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

atomicidad, consistencia, aislamiento, durabilidad (ACID)

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

control de acceso basado en atributos (ABAC)

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

origen de datos fidedigno

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

Zona de disponibilidad

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

AWS Marco de adopción de la nube (AWS CAF)

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

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

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

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

B

bot malicioso

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

BCP

Consulte planificación de la continuidad del negocio.

gráfico de comportamiento

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

sistema big-endian

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

clasificación binaria

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

filtro de floración

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

implementación azul/verde

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

bot

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

botnet

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

branch

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

acceso de emergencia

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

estrategia de implementación sobre infraestructura existente

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

caché de búfer

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

capacidad empresarial

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

planificación de la continuidad del negocio (BCP)

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

C

CAF

Consulte AWS Cloud Adoption Framework.

implementación canario

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

CCoE

Consulte Centro de excelencia en la nube.

CDC

Consulte captura de datos de cambios.

captura de datos de cambio (CDC)

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

ingeniería del caos

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

CI/CD

Consulte integración continua y entrega continua.

clasificación

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

cifrado del cliente

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

Centro de excelencia en la nube (CCoE)

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

computación en la nube

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

modelo operativo en la nube

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

etapas de adopción de la nube

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

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

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

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

CMDB

Consulte base de datos de administración de configuración.

repositorio de código

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

caché en frío

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

datos fríos

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

visión artificial (CV)

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

deriva de configuración

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

base de datos de administración de configuración (CMDB)

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

paquete de conformidad

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

integración y entrega continuas (CI/CD)

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

CV

Consulte visión artificial.

D

datos en reposo

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

clasificación de datos

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

deriva de datos

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

datos en tránsito

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

malla de datos

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

minimización de datos

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

perímetro de datos

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

preprocesamiento de datos

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

procedencia de los datos

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

titular de los datos

Persona cuyos datos se recopilan y procesan.

almacenamiento de datos

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

lenguaje de definición de datos (DDL)

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

lenguaje de manipulación de datos (DML)

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

DDL

Consulte lenguaje de definición de bases de datos.

conjunto profundo

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

aprendizaje profundo

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

defense-in-depth

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

administrador delegado

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

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

Implementación

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

entorno de desarrollo

Consulte entorno.

control de detección

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

asignación de flujos de valor para el desarrollo (DVSM)

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

gemelo digital

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

tabla de dimensiones

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

desastre

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

recuperación de desastres (DR)

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

DML

Consulte lenguaje de manipulación de bases de datos.

diseño basado en el dominio

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

DR

Consulte recuperación ante desastres.

Detección de desviaciones

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

DVSM

Consulte asignación de flujos de valor para el desarrollo.

E

EDA

Consulte análisis de datos de tipo exploratorio.

EDI

Consulte intercambio electrónico de datos.

computación en la periferia

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

intercambio electrónico de datos (EDI)

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

cifrado

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

clave de cifrado

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

endianidad

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

punto de conexión

Consulte punto de conexión de servicio.

servicio de punto de conexión

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

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

planificación de recursos empresariales (ERP)

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

cifrado de sobre

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

entorno

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

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

epopeya

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

ERP

Consulte planificación de recursos empresariales.

análisis de datos de tipo exploratorio (EDA)

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

F

tabla de hechos

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

Fail Fast

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

límite de aislamiento de errores

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

rama de característica

Consulte rama.

características

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

importancia de las características

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

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

transformación de funciones

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

peticiones con pocos pasos

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

FGAC

Consulte control de acceso detallado.

control de acceso preciso (FGAC)

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

migración relámpago

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

FM

Consulte modelo fundacional.

Modelo fundacional (FM)

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

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

G

IA generativa

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

bloqueo geográfico

Consulte restricciones geográficas.

restricciones geográficas (bloqueo geográfico)

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

Flujo de trabajo de Gitflow

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

imagen dorada

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

estrategia de implementación desde cero

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

barrera de protección

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

H

HA

Consulte alta disponibilidad.

migración heterogénea de bases de datos

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

alta disponibilidad (HA)

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

modernización histórica

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

datos de reserva

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

migración homogénea de bases de datos

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

datos recientes

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

hotfix

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

periodo de hiperatención

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

I

IaC

Consulte infraestructura como código.

políticas basadas en identidades

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

aplicación inactiva

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

IIoT

Consulte Internet de las cosas industrial.

infraestructura inmutable

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

VPC entrante (de entrada)

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

migración gradual

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

Industria 4.0

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

infraestructura

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

infraestructura como código (IaC)

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

Internet de las cosas industrial (T) IIo

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

VPC de inspección

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

Internet de las cosas (IoT)

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

interpretabilidad

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

IoT

Consulte Internet de las cosas.

biblioteca de información de TI (ITIL)

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

administración de servicios de TI (ITSM)

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

ITIL

Consulte biblioteca de información de TI.

ITSM

Consulte administración de servicios de TI.

L

control de acceso basado en etiquetas (LBAC)

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

zona de aterrizaje

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

modelo de lenguaje de gran tamaño (LLM)

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

migración grande

Migración de 300 servidores o más.

LBAC

Consulte control de acceso basado en etiquetas.

privilegio mínimo

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

migrar mediante lift-and-shift

Consulte Las 7 R.

sistema little-endian

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

LLM

Consulte modelo de lenguaje de gran tamaño.

entornos inferiores

Consulte entorno.

M

machine learning (ML)

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

rama principal

Consulte rama.

malware

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

Servicios administrados

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

sistema de ejecución de fabricación (MES)

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

MAP

Consulte Programa de aceleración de la migración.

mecanismo

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

cuenta de miembro

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

MES

Consulte sistema de ejecución de fabricación.

Message Queuing Telemetry Transport (MQTT)

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

microservicio

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

arquitectura de microservicios

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

Programa de aceleración de la migración (MAP)

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

migración a escala

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

fábrica de migración

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

metadatos de migración

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

patrón de migración

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

Migration Portfolio Assessment (MPA)

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

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

Evaluación de la preparación para la migración (MRA)

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

estrategia de migración

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

ML

Consulte machine learning.

modernización

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

evaluación de la preparación para la modernización

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

aplicaciones monolíticas (monolitos)

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

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

MPA

Consulte Migration Portfolio Assessment.

MQTT

Consulte Message Queuing Telemetry Transport.

clasificación multiclase

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

infraestructura mutable

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

O

OAC

Consulte control de acceso de origen.

OAI

Consulte identidad de acceso de origen.

OCM

Consulte administración del cambio organizacional.

migración fuera de línea

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

OI

Consulte integración de operaciones.

OLA

Consulte acuerdo de nivel operativo.

migración en línea

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

OPC-UA

Consulte Open Process Communications: arquitectura unificada.

Open Process Communications: arquitectura unificada (OPC-UA)

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

acuerdo de nivel operativo (OLA)

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

revisión de la preparación operativa (ORR)

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

tecnología operativa (TO)

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

integración de operaciones (OI)

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

registro de seguimiento organizativo

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

administración del cambio organizacional (OCM)

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

control de acceso de origen (OAC)

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

identidad de acceso de origen (OAI)

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

ORR

Consulte revisión de la preparación operativa.

OT

Consulte tecnología operativa.

VPC saliente (de salida)

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

P

límite de permisos

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

información de identificación personal (PII)

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

PII

Consulte información de identificación personal.

manual de estrategias

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

PLC

Consulte controlador lógico programable.

PLM

Consulte administración del ciclo de vida del producto.

policy

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

persistencia políglota

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

evaluación de cartera

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

predicate

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

inserción de predicados

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

control preventivo

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

entidad principal

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

Privacidad desde el diseño

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

zonas alojadas privadas

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

control proactivo

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

administración del ciclo de vida del producto (PLM)

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

entorno de producción

Consulte entorno.

controlador lógico programable (PLC)

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

encadenamiento de peticiones

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

seudonimización

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

publish/subscribe (pub/sub)

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

Q

plan de consulta

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

regresión del plan de consulta

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

R

Matriz RACI

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

RAG

Consulte generación aumentada por recuperación.

ransomware

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

Matriz RASCI

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

RCAC

Consulte control de acceso por filas y columnas.

réplica de lectura

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

rediseñar

Consulte Las 7 R.

objetivo de punto de recuperación (RPO)

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

objetivo de tiempo de recuperación (RTO)

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

refactorizar

Consulte Las 7 R.

Region

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

regresión

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

volver a alojar

Consulte Las 7 R.

versión

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

reubicar

Consulte Las 7 R.

redefinir la plataforma

Consulte Las 7 R.

recomprar

Consulte Las 7 R.

resiliencia

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

política basada en recursos

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

matriz responsable, confiable, consultada e informada (RACI)

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

control receptivo

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

retain

Consulte Las 7 R.

retirar

Consulte Las 7 R.

Generación aumentada de recuperación (RAG)

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

rotación

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

control de acceso por filas y columnas (RCAC)

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

RPO

Consulte objetivo de punto de recuperación.

RTO

Consulte objetivo de tiempo de recuperación.

manual de procedimientos

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

S

SAML 2.0

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

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

SCADA

Consulte control de supervisión y adquisición de datos.

SCP

Consulte política de control de servicio.

secreta

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

seguridad desde el diseño

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

control de seguridad

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

refuerzo de la seguridad

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

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

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

automatización de la respuesta de seguridad

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

cifrado del servidor

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

política de control de servicio (SCP)

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

punto de enlace de servicio

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

acuerdo de nivel de servicio (SLA)

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

indicador de nivel de servicio (SLI)

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

objetivo de nivel de servicio (SLO)

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

modelo de responsabilidad compartida

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

SIEM

Consulte sistema de administración de eventos e información de seguridad.

único punto de error (SPOF)

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

SLA

Consulte acuerdo de nivel de servicio.

SLI

Consulte indicador de nivel de servicio.

SLO

Consulte objetivo de nivel de servicio.

split-and-seed modelo

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

SPOF

Consulte único punto de error.

esquema en estrella

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

patrón de higo estrangulador

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

subred

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

control de supervisión y adquisición de datos (SCADA)

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

cifrado simétrico

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

pruebas sintéticas

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

petición del sistema

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

T

etiquetas

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

variable de destino

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

lista de tareas

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

entorno de prueba

Consulte entorno.

entrenamiento

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

puerta de enlace de tránsito

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

flujo de trabajo basado en enlaces troncales

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

acceso de confianza

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

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

ajuste

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

equipo de dos pizzas

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

U

incertidumbre

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

tareas indiferenciadas

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

entornos superiores

Consulte entorno.

V

succión

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

control de versión

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

Emparejamiento de VPC

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

vulnerabilidad

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

W

caché caliente

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

datos templados

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

función de ventana

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

carga de trabajo

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

flujo de trabajo

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

WORM

Consulte escritura única y lectura múltiple.

WQF

Consulte AWS Workload Qualification Framework.

escritura única y lectura múltiple (WORM)

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

Z

ataque de día cero

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

vulnerabilidad de día cero

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

peticiones desde cero

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

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

aplicación zombi

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

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