Aumentar la resiliencia y mejorar la experiencia del cliente mediante el uso de la ingeniería del caos en AWS

# AWS Orientación prescriptiva

# AWS Orientación prescriptiva: Aumentar la resiliencia y mejorar la experiencia del cliente mediante el uso de la ingeniería del caos en AWS

# Table of Contents

# Aumentar la resiliencia y mejorar la experiencia del cliente mediante el uso de la ingeniería del caos en AWS

Laurent Domb, tecnólogo jefe de Finanzas Federales de Amazon Web Services

Abril de 2025 (historial del documento)

La ingeniería del caos es la disciplina que consiste en experimentar con una aplicación para generar confianza en la capacidad de la organización y de la aplicación para soportar condiciones turbulentas de producción. Se trata de un enfoque proactivo de la resiliencia, cuyo objetivo es comprobar si la aplicación y la organización son capaces de absorber las deficiencias del servicio, adaptarse a ellas y, en última instancia, recuperarse de ellas, mediante la introducción de fallos controlados en las personas, los procesos y la tecnología. La intención también es identificar y eliminar las debilidades antes de que puedan provocar cortes u otras interrupciones en la producción.

En Amazon, entendemos que los fallos son inevitables en los sistemas distribuidos, hasta el punto de que funcionar a pesar de la presencia de fallos es un modo de funcionamiento normal. Dado que las interacciones entre los servicios están destinadas a fallar, debe comprender cómo reaccionan sus servicios ante los distintos modos de fallo y crear servicios que sean resistentes a las principales vulnerabilidades, como los fallos de dependencia, las tormentas de reintentos, las zonas de disponibilidad deterioradas y el agotamiento de los recursos del host.

Tomemos el ejemplo de una tormenta de reintentos. Un fallo localizado en un cliente puede afectar significativamente a varios servicios. Esto se conoce comúnmente como efecto mariposa. Una tormenta de reintentos es una manifestación del efecto mariposa, en el que una dependencia fallida hace que los clientes, y los clientes de esos clientes, vuelvan a intentar la operación fallida, lo que se traduce en un crecimiento exponencial del tráfico. Los servicios se sobrecargan porque deben responder al tráfico normal además de volver a intentarlo y, al mismo tiempo, gestionar una degradación del rendimiento.

La ingeniería del caos ha surgido como respuesta a la creciente complejidad de los sistemas distribuidos. Es un enfoque multidisciplinario que combina principios de la teoría del caos, el pensamiento sistémico y la ingeniería para diseñar y administrar sistemas complejos que sean resistentes a eventos y comportamientos inesperados. En esencia, la ingeniería del caos se ocupa de comprender y gestionar el comportamiento de los sistemas complejos en condiciones de incertidumbre e imprevisibilidad. Reconoce que los enfoques tradicionales de la ingeniería, que se basan en la predicción y el control de los resultados, suelen ser insuficientes para hacer frente a la

naturaleza compleja y dinámica de los sistemas distribuidos. A medida que estos sistemas crecen, suelen superar el alcance de comprensión de cualquier individuo.

La ingeniería del caos proporciona conceptos, técnicas y herramientas para introducir errores intencionalmente en los sistemas y descubrir las debilidades antes de que se manifiesten en la producción. Este enfoque proactivo permite a las organizaciones generar confianza en que sus sistemas funcionarán en condiciones estresantes. Si bien la ingeniería del caos sigue siendo una práctica en evolución, representa un cambio fundamental hacia el diseño, la gestión y el funcionamiento de los sistemas informáticos modernos para que sean resilientes ante el aumento de la complejidad y la interconexión.

En las siguientes secciones de esta guía se analizan las ventajas de la ingeniería del caos, se explica cómo realizar experimentos de ingeniería del caos y se describen los enfoques que puede adoptar para implementar la ingeniería del caos a escala en su organización. También se incluyen ejemplos de documentos sobre la planificación de los experimentos y los resultados de los experimentos que puede utilizar como plantillas para sus experimentos de ingeniería del caos.



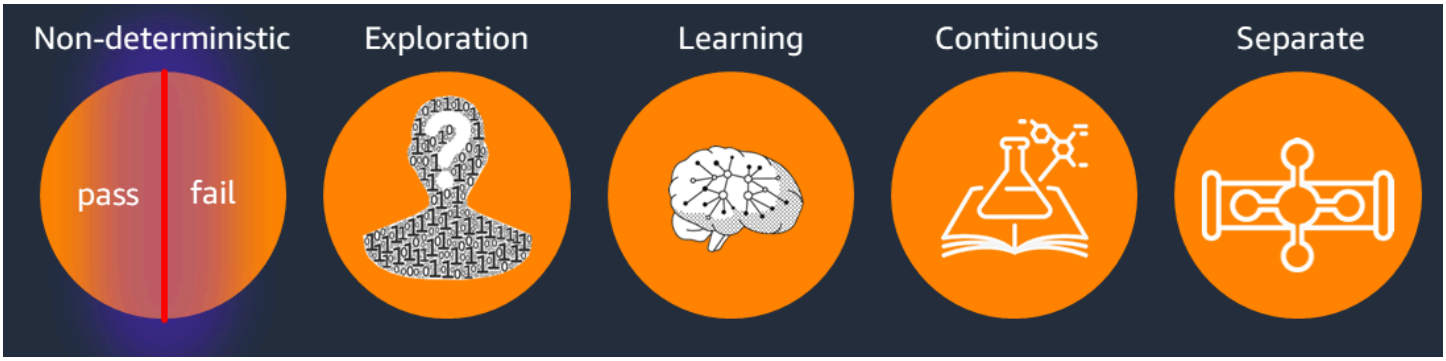
La siguiente sección analiza en qué se diferencian las características de la ingeniería del caos de las pruebas de resiliencia tradicionales, como las pruebas unitarias, de humo o de integración.

# Descripción general de

## Comparación de las pruebas de resiliencia con la ingeniería del caos

Las pruebas de resiliencia son deterministas. Es decir, validan las características conocidas sobre los mecanismos de resiliencia, como los disyuntores, los reintentos, las conmutaciones por error o las alternativas, que se han implementado en la aplicación. Confirma la forma en que estos componentes de la aplicación absorben las interrupciones controladas con un impacto mínimo o nulo para el usuario. Por lo tanto, las pruebas de resiliencia se centran en la validación de los modos de fallo conocidos que se incorporan a los componentes de la aplicación con el objetivo de producir pass/fail resultados. Debe realizar pruebas de resiliencia de forma continua como parte de su proceso para asegurarse de no introducir regresiones en su postura de resiliencia. En las pruebas de resiliencia, a menudo no se realizan pruebas con componentes reales, sino que se simulan API que simulan un componente determinado. Este enfoque permite realizar pruebas consistentes y reproducibles de los escenarios de falla en un entorno controlado, lo que lo hace adecuado para la integración automatizada de canalizaciones y las pruebas de regresión.


Por el contrario, la ingeniería del caos no es determinista. Es decir, se basa en hipótesis y verifica tu modelo mental sobre la forma en que la aplicación y sus dependencias (personas, procesos y tecnología) absorben los modos de fallo imprevistos, se adaptan a ellos y, finalmente, se recuperan de ellos. Por lo tanto, la ingeniería del caos se centra en la verificación integral de los modos de falla desconocidos, con el objetivo de detectar los defectos a tiempo y subsanarlos antes de que se conviertan en eventos a gran escala. La ingeniería del caos fomenta el aprendizaje continuo y debe practicarse mediante un proceso de creación de caos independiente o mediante experimentos ad hoc que permitan ejecutar varios experimentos en cualquier momento sin bloquear la productividad del desarrollador a la hora de implementar el código.




El proceso de ingeniería del caos suele comenzar con un día de juego caótico, que es un evento específico en el que los equipos introducen intencionadamente fallos o fallos controlados en su aplicación. El día de juego es progresivo: comienza en entornos de nivel inferior (como el desarrollo o las pruebas) y, poco a poco, avanza hacia entornos de nivel superior (como la puesta en escena y la preproducción) a medida que aumenta la confianza. Al moverse sistemáticamente por estos entornos, los equipos pueden comprobar que sus sistemas toleran adecuadamente los fallos inyectados antes de que lleguen a la fase de producción. Esta progresión metódica garantiza que, cuando se lleven a cabo experimentos de caos en entornos de producción, los equipos hayan adquirido una confianza sustancial en las capacidades de resiliencia de sus sistemas. El proceso del día del partido es un enfoque proactivo para identificar las debilidades y vulnerabilidades en la arquitectura y las prácticas operativas de una aplicación y, al mismo tiempo, eliminar el stress de aprender durante una interrupción inesperada de la producción.

## El valor de la ingeniería del caos

Los sistemas complejos son omnipresentes en el mundo actual. Desempeñan un papel fundamental en muchos aspectos de nuestras vidas, desde los mercados financieros hasta la sanidad. Esperamos que estos sistemas estén siempre operativos. Sin embargo, los sistemas complejos suelen ser vulnerables a eventos y comportamientos inesperados que pueden tener consecuencias importantes. Las organizaciones deben planificar para la disrupción en lugar de preguntarse si ocurrirá. Para ello, pueden aplicar pruebas de escenarios en todos sus servicios empresariales críticos o de misión crítica. Aquí es donde entra en juego la ingeniería del caos.

La ingeniería del caos ofrece un enfoque para gestionar sistemas complejos que puede ayudar a mitigar los riesgos y mejorar la resiliencia. El proceso de preparación para los experimentos de caos requiere que los equipos desarrollen hipótesis sobre el comportamiento de sus sistemas, lo que les permite comprender mejor cómo se construyen los sistemas y cómo funcionan. Esta preparación suele revelar lagunas mentales, ideas arquitectónicas y conocimientos operativos que, de otro modo,

podrían quedar sin descubrir. Al comprender mejor cómo reaccionan los sistemas complejos ante las fallas, la ingeniería del caos promueve una mayor transparencia y responsabilidad en el diseño y la administración de los sistemas. Cuanto más a menudo su organización practique la ingeniería del caos, mejor preparada estará desde el punto de vista operativo. La ingeniería del caos le ayuda a establecer las mejores prácticas para diseñar aplicaciones resilientes que puedan sobrevivir a los fallos de los componentes con un impacto mínimo o nulo en los usuarios. Esto garantiza que las aplicaciones críticas funcionen dentro de los niveles de servicio y la tolerancia a los impactos esperados, al tiempo que mejora continuamente el conocimiento del equipo sobre sus propios sistemas.

## Prepararse para condiciones adversas

Al desarrollar AWS, utilizas distintos tipos de servicios, incluidos servicios zonales como Amazon Elastic Compute Cloud (Amazon EC2), servicios regionales como Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), servicios globales como AWS Identity and Access Management (IAM), servicios de software como servicio (SaaS) de terceros y servicios locales. Cada tipo de servicio expone distintos dominios de error que hay que tener en cuenta. ¿Cómo se prepara para los eventos autoinfligidos o los eventos causados por terceros sobre los que su organización no tiene control?

Para ayudarte a entender cómo podría responder tu aplicación a condiciones adversas, puedes usar AWS Fault Injection Service ()AWS FIS. AWS FIS es un servicio totalmente gestionado para ejecutar experimentos de inyección de fallos de forma controlada. Puede utilizar este servicio para detectar situaciones AWS predeterminadas, como interrupciones del suministro eléctrico en la zona de disponibilidad o problemas de conectividad entre regiones, o bien crear sus propios experimentos combinando una amplia variedad de errores que ofrece el servicio. AWS FIS permite a sus equipos practicar y aprender continuamente cómo reaccionaría su aplicación ante los fallos más comunes y cómo remediaría los defectos a medida que los detecten.

## Practicando la ingeniería del caos controlada

Los principios clave de los experimentos de caos controlado son:

- Comience con un entorno similar a su entorno de producción.
- Establezca una hipótesis y condiciones de parada para su experimento.
- Empezar poco a poco.

- Controla tus experimentos de caos.
- Establece el alcance del impacto.
- Conozca la línea base de su servicio.
- Programa los experimentos.
- Remedie primero y luego experimente.
- Supervisa el experimento de cerca.
- Aprenda de sus resultados.
- Priorice los hallazgos, corrija y verifique.
- Propague los aprendizajes en toda su organización.

Para escalar con éxito la ingeniería del caos, debe implementar experimentos de caos de forma controlada. Cuando lo usas AWS FIS, puedes crear condiciones de parada mediante las CloudWatchalarmas de Amazon. Puede incorporar estas condiciones en una plantilla de experimento para asegurarse de que los experimentos se detengan si se sobrepasan los límites y vuelvan a su último estado conocido. AWS FIS también proporciona palancas de seguridad. Al activar estas palancas, AWS FIS se detienen y anulan todos los experimentos que se estén realizando en la cuenta Región de AWS, incluidos los experimentos con varias cuentas, e impide que se inicien nuevos experimentos. Esto evita que se produzcan errores durante determinados períodos de tiempo, como el horario de negociación, los eventos de rebajas o el lanzamiento de productos, o en respuesta a las alarmas de estado de las aplicaciones. La palanca de seguridad permanece activada hasta que se desactiva manualmente.

Cuando lleves a cabo un experimento caótico, debes definir medidas de seguridad para evitar efectos secundarios no deseados en el medio ambiente, especialmente si existe la posibilidad de que el experimento afecte a las aplicaciones que están en producción. Cuando planifique el experimento, prevea cualquier efecto adverso que pueda tener en otras aplicaciones del entorno. Por ejemplo, otras aplicaciones podrían recibir mensajes erróneos de la aplicación que forma parte del experimento, recibir un gran volumen de solicitudes o encontrarse con problemas de recursos si comparten la infraestructura. Documente estos riesgos y aborde cualquier problema conocido o inaceptable antes de ejecutar el experimento.

# Cómo empezar con la ingeniería del caos

Antes de realizar un experimento, le recomendamos que ponga en práctica algunos elementos esenciales para aprovechar al máximo sus prácticas de ingeniería del caos. Estos elementos esenciales incluyen:

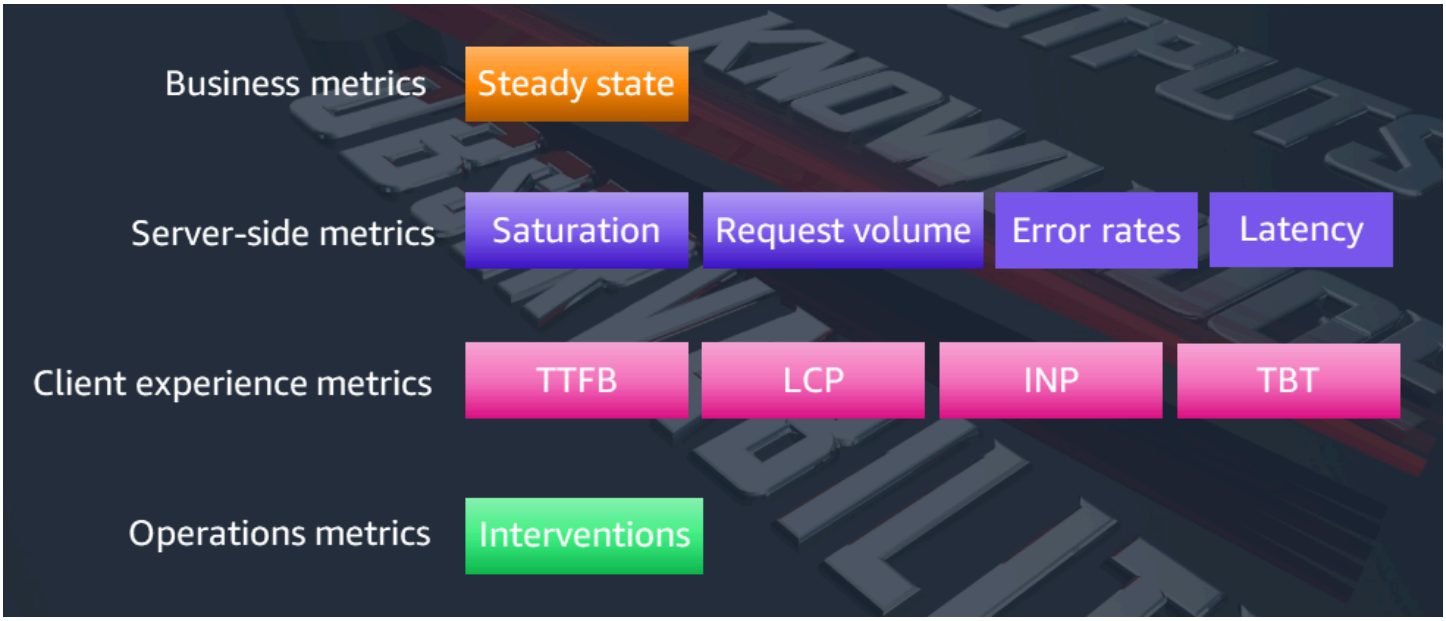
- Observabilidad (métricas, registro, seguimiento de solicitudes)
- Una lista de eventos o fallas del mundo real que te gustaría explorar
- Patrocinio de la resiliencia organizacional mediante la participación de los líderes
- Priorización de los hallazgos críticos, en función del posible impacto empresarial, por encima de las nuevas características que se descubran al realizar experimentos de caos

# Observabilidad para los experimentos de caos

La observabilidad, que comprende las métricas, el registro y el seguimiento de las solicitudes, desempeña un papel clave en la ingeniería del caos. Cuando realice un experimento, querrá comprender las métricas empresariales, las métricas del lado del servidor, las métricas de la experiencia del cliente y las métricas de las operaciones. Sin la observabilidad, no podrá definir el comportamiento en estado estacionario ni crear un experimento significativo para verificar si su hipótesis sobre su aplicación es cierta.

## Métricas

El siguiente diagrama muestra los tipos de métricas de las que puede hacer un seguimiento para realizar experimentos de caos en distintos tipos de aplicaciones.




- Métricas empresariales: el estado estable indica el funcionamiento normal del sistema y lo definen las métricas empresariales. Se puede representar mediante transacciones por segundo (TPS), flujos de clics por segundo, pedidos por segundo o una medición similar. La aplicación muestra un estado estable cuando funciona según lo esperado. Por lo tanto, compruebe que la aplicación esté en buen estado antes de ejecutar los experimentos. El estado estable no significa necesariamente que no se produzca ningún impacto en la aplicación cuando se produzca un error, ya que un porcentaje de errores podría estar dentro de los límites aceptables. El estado estacionario es su punto de referencia. Por ejemplo, el estado estable de un sistema de pagos podría definirse como el procesamiento de 300 TPS con una tasa de éxito del 99 por ciento y un tiempo de ida y vuelta de 500 ms. Visualmente, piense en el estado estacionario como un electrocardiograma (ECG). Si el estado estable de su sistema fluctúa repentinamente, sabrá que hay un problema con el servicio.
- Server-side métricas: para entender el rendimiento de tus recursos durante el experimento, necesitas información sobre su rendimiento antes, durante y después del experimento. Para medir el impacto de tus recursos en AWS, puedes usar Amazon CloudWatch. CloudWatch es un servicio que monitorea las aplicaciones, responde a los cambios de rendimiento, optimiza el uso de los recursos y proporciona información sobre el estado de las operaciones. Durante sus experimentos, querrá capturar métricas del lado del servidor, como la saturación, los volúmenes de solicitudes, las tasas de error y la latencia.
- Métricas de experiencia del cliente: sí AWS, puedes capturar métricas de usuarios reales mediante CloudWatchRUM para simular las solicitudes de los usuarios a través de herramientas como Locust, Grafana k6, Selenium o Puppeteer. Las métricas reales de los usuarios son

fundamentales para las organizaciones que llevan a cabo experimentos de ingeniería del caos. Al monitorear cómo se ven afectados los usuarios reales durante un experimento, los equipos pueden obtener una imagen precisa de cómo los fallos y las interrupciones afectarán a los clientes en la producción. Los indicadores clave de la experiencia del cliente son el tiempo hasta el primer byte (TTFB), Largest Contentful Paint (LCP), la interacción con el siguiente cuadro (INP) y el tiempo total de bloqueo (TBT).

- Métricas operativas: las intervenciones miden la eficacia con la que se mitigan los fallos de forma automatizada; por ejemplo, la latencia correcta de las solicitudes de los clientes durante el reinicio de módulos, tareas o instancias de EC2 con mecanismos como el controlador de replicación o el escalado automático. Poder intervenir automáticamente durante una avería está directamente relacionado con una buena experiencia de usuario. Es crucial saber si estos mecanismos de mitigación se ven alterados a lo largo del tiempo. Al definir las métricas para las mitigaciones automatizadas exitosas y fallidas, se crean pautas que ayudan a identificar las posibles regresiones en todo el sistema.

### Registro

El registro centralizado es fundamental para comprender los estados de los componentes de la aplicación antes, durante y después de un experimento caótico. Le recomendamos que recopile registros de todos los componentes de la aplicación para crear una vista consolidada de lo que hacía cada componente en el momento en que se introdujo el experimento. Esto proporciona una imagen clara del flujo del experimento de principio a fin.

### Rastreo de solicitudes

El seguimiento de las solicitudes le permite observar el flujo de cualquier solicitud individual en todos los componentes de la aplicación para obtener una comprensión completa del impacto que la falla inyectada tiene en el sistema y sus dependencias. Al rastrear las solicitudes, puede ver cómo se propaga la falla a través de los diferentes servicios y componentes, lo que le permite evaluar mejor el alcance de la interrupción. Para realizar un seguimiento de sus solicitudes AWS, puede utilizar. AWS X-Ray

## Escenarios de fracaso para provocar el caos en los experimentos

El objetivo de introducir errores comunes en su aplicación es comprender cómo reacciona la aplicación ante estos eventos inesperados, de modo que pueda crear mecanismos de mitigación y

hacer que su sistema sea resistente a dichos errores. Además, debe utilizar la ingeniería del caos para reproducir los escenarios de fallos históricos y comprobar que sus mecanismos de mitigación siguen funcionando según lo previsto y que no se han desviado con el paso del tiempo.

Ten en cuenta los siguientes eventos cuando planifiques tus experimentos de ingeniería del caos.

| Modo de fallo | Description (Descripción) |
| --- | --- |
| Deterioro del servidor | Reinicie las instancias EC2, elimine los pods de Kubernetes o las tareas de Amazon Elastic Container Service (Amazon ECS) para comprender cómo reacciona su aplicación ante estos bloqueos. |
| errores de API | Introduzca errores en las API de servicio AWS y en sus propias API de servicio para comprender el comportamiento de las aplicaciones. |
| Problemas de red | Introduzca latencia o congestión, o bloquee las conexiones para imitar los problemas de red del mundo real. |
| AWS Deterioro de zona de disponibilidad | Reproduzca el deterioro de una zona de disponibilidad completa para verificar la recuperación en todas las zonas. |
| Región de AWS deterioro de la conectividad | Reproduzca una avería de la red Regiones de AWS para comprobar cómo reaccionan los recursos de la región secundaria ante tal suceso. |
| Fallos en la base | Conmute por error las réplicas de las bases de datos o dañe los datos, o impida el acceso a las instancias de la base de datos, para comprender el impacto en sus estrategias de aplicaciones y recuperación. |

| Modo de fallo | Description (Descripción) |
| --- | --- |
| Pausa en la replicación de la base de datos y Amazon S3 | Detenga la replicación de la base de datos o de Amazon Simple Storage Service (Amazon S3) en todas las zonas de disponibilidad Regiones de AWS o para comprender el impacto en las aplicaciones posteriores. |
| Degradación del almacenamiento | Pausa I/O, elimina los volúmenes de Amazon Elastic Block Store (Amazon EBS) o elimina archivos para comprobar la durabilidad y la recuperación de los datos. |
| Deterioro de dependencia | Reduzca o reduzca el rendimiento de los servicios descendentes o ascendentes de los que depende, incluidos los servicios de terceros, para comprender el flujo de principio a fin y el impacto en sus clientes. |
| Aumentos repentinos de tráfico | Genere picos en el tráfico de usuarios para probar las capacidades de escalado automátic o y compruebe cómo el tiempo de arranque en frío puede afectar al estado general de la aplicación. |
| Agotamiento de los recursos | Aproveche al máximo la CPU, la memoria y el espacio en disco para comprobar la correcta degradación de la aplicación. |
| Fallos en cascada | Provocan los errores principales que repercute n en cascada en las aplicaciones y los componentes posteriores. |
| Implementaciones incorrectas | Implemente cambios o configuraciones problemáticos para verificar los mecanismos de reversión. |

# Patrocinio de la resiliencia organizacional

La ingeniería del caos proporciona el mayor valor cuando se aplica en toda la organización. Le recomendamos que trabaje con un patrocinador ejecutivo que pueda ayudarlo a establecer objetivos de resiliencia en toda su organización, eliminar el miedo, la incertidumbre y las dudas sobre el ámbito e iniciar el proceso de transformación para que la resiliencia sea responsabilidad de todos.

Para respaldar el argumento empresarial de crear una práctica de ingeniería basada en el caos, incorpore las iniciativas de ingeniería del caos a sus proyectos empresariales fundamentales. Hacer de la resiliencia un activo y un motor de aceleración le ayudará a medir el éxito a lo largo del tiempo. Comience con un recuento de los incidentes críticos por mes o por trimestre, el tiempo promedio de recuperación y el impacto que estos incidentes causaron a sus clientes y a su organización. Establezca un objetivo con sus equipos para reducir el número de incidentes en un período de 6 a 12 meses, a medida que se vayan introduciendo mejoras en todos sus paquetes de aplicaciones en respuesta a los complicados experimentos de ingeniería.

Mida si los incidentes son muy repetitivos. Por ejemplo, supongamos que un certificado TLS caducado provoca un tiempo de inactividad porque los clientes no pueden establecer una conexión de confianza. Si se producen varios incidentes en un año debido a la caducidad de varios certificados TLS, puedes realizar un experimento sobre la caducidad de un certificado TLS y comprobar que tus equipos reciben alertas o son capaces de mitigar automáticamente dichos problemas. Esto ayudará a garantizar que seas resistente a este tipo de errores.

Para hacer un seguimiento del progreso de la ingeniería del caos a lo largo del tiempo, registre las siguientes métricas para ayudar a resaltar el valor de la ingeniería del caos a lo largo del ciclo de vida de una aplicación:

- Menor tasa de incidentes: haga un seguimiento del número de incidentes de producción a lo largo del tiempo y correlacione este número con la adopción de la ingeniería del caos. La expectativa es que la tasa de incidentes graves disminuya.
- Mejora del tiempo medio de resolución (MTTR): calcule el tiempo medio que se tarda en resolver los incidentes y realice un seguimiento de estos datos para comprobar si mejoran con la ingeniería del caos a lo largo del tiempo.
- Mayor disponibilidad de las aplicaciones: utilice métricas de nivel de servicio para mostrar las mejoras de disponibilidad a medida que aumenta la resiliencia de las aplicaciones debido a los experimentos caóticos.

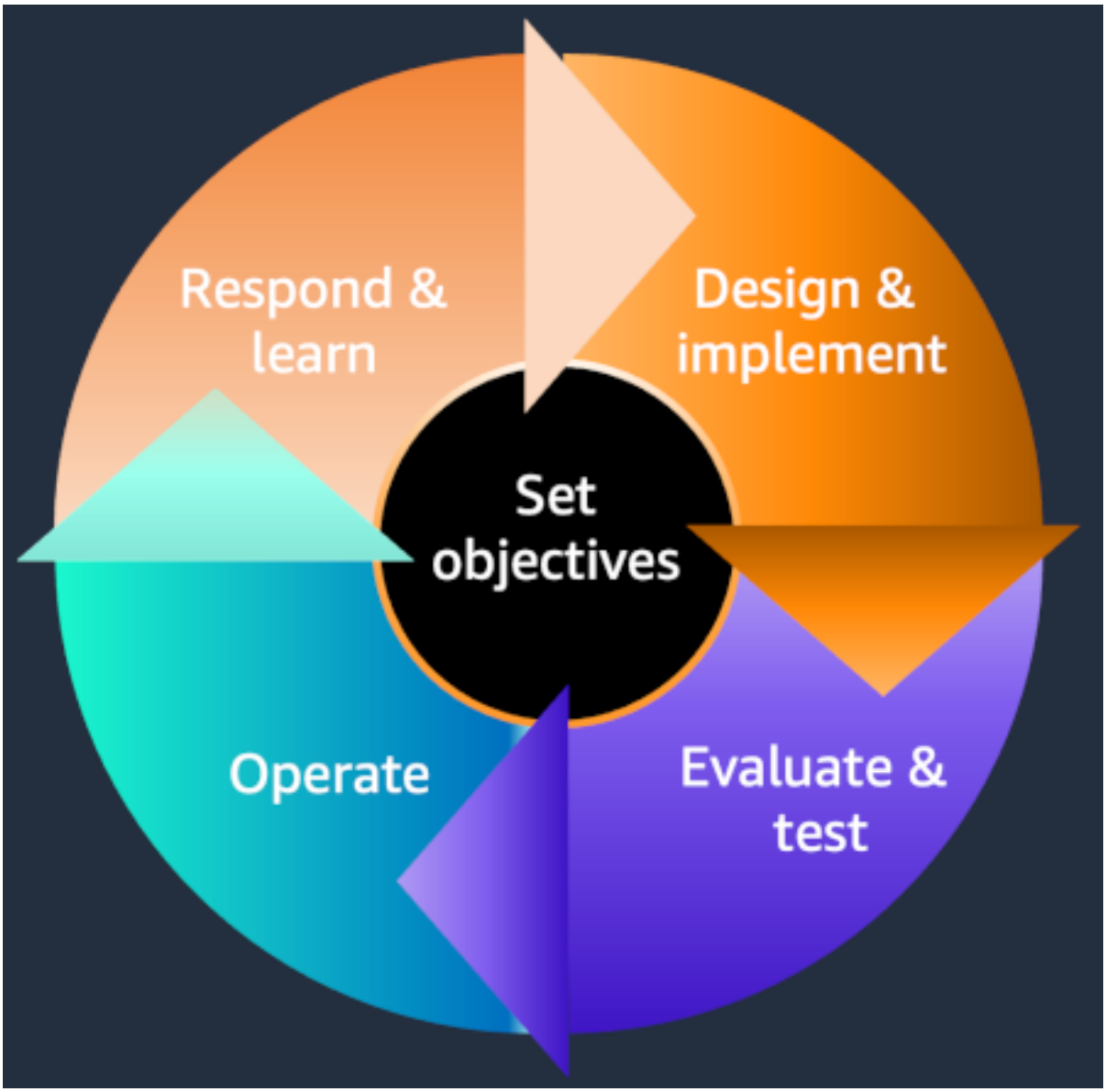
- Tiempo de comercialización más rápido: la ingeniería basada en el caos puede proporcionarle la confianza necesaria para lanzar ofertas innovadoras con mayor rapidez, ya que sabe que sus aplicaciones son resilientes. Realice un seguimiento de los aumentos en la velocidad de lanzamiento de los productos.
- Reducción de los costos operativos: cuantifique si los indicadores como el ruido de las alertas, la carga operativa y el esfuerzo manual para administrar las aplicaciones disminuyen debido a las prácticas caóticas.
- Aumentar la confianza: encueste a desarrolladores, ingenieros de confiabilidad de sitios (SRE) y otro personal técnico para evaluar si la ingeniería del caos aumentó su confianza en la resiliencia de las aplicaciones. Las percepciones importan.
- Experiencias de cliente mejoradas: Connect la ingeniería del caos con resultados positivos para los clientes, como menos interrupciones del servicio, retrocesos e interrupciones.

## Priorizar la remediación

Al realizar experimentos de caos, es probable que identifique áreas de mejora en las que la aplicación no funciona según lo previsto. La solución de estos problemas se convertirá en una tarea pendiente que tendrá que priorizar junto con otras tareas, como el desarrollo de funciones. Le recomendamos que dedique tiempo a estas mejoras para evitar futuros errores. Considere la posibilidad de priorizar estos aprendizajes y tareas de corrección en función del nivel de impacto que puedan causar. Los hallazgos que afecten directamente a la resiliencia o la seguridad de la aplicación deberían tener prioridad sobre las nuevas funciones, a fin de evitar que repercutan en los clientes. Si el equipo se esfuerza por priorizar las tareas de remediación antes que el desarrollo de funciones, considere la posibilidad de ponerse en contacto con su patrocinador ejecutivo para asegurarse de que las prioridades se establezcan en función de la tolerancia al riesgo empresarial.

# Implementación de la ingeniería del caos en AWS

La ingeniería del caos forma parte de la etapa de evaluación y prueba del ciclo de vida de la AWS resiliencia, como se ilustra en el siguiente diagrama. Las aplicaciones distribuidas no funcionan de forma aislada de otras aplicaciones o clientes, por lo que le recomendamos que revise todo el ciclo de vida de la resiliencia. El cambio es constante en las aplicaciones distribuidas a medida que la red evoluciona, las aplicaciones ascendentes y descendentes sufren cambios y el uso de los clientes cambia con el tiempo.



Para comprender cómo estos cambios en su aplicación pueden afectar a su capacidad de recuperación, incorpore la ingeniería del caos a sus operaciones diarias. Puede implementar experimentos de caos de diferentes maneras:

- Ad hoc: puedes realizar experimentos de caos como experimentos únicos para abordar un problema o una pregunta específicos.
- Días de juego caóticos: se trata de eventos estructurados y recurrentes diseñados para comprobar la fiabilidad y la resiliencia de una aplicación. El objetivo de un día de juego caótico es identificar posibles problemas o deficiencias de resiliencia en las personas, los procesos y las tecnologías, y poner en práctica los procesos y procedimientos para identificar, mitigar y responder a los incidentes.
- Chaos Pipeline: la integración y la entrega continuas (CI/CD) consisten en crear nuevas funciones e implementarlas de forma segura en todos los entornos. Para implementar experimentos de ingeniería del caos, crea una canalización del caos que esté separada de la CI/CD tuya. Para entender por qué, supongamos que quieres añadir a tu CI/CD proceso un único experimento de ingeniería del caos que provoque una pérdida de paquetes cada vez mayor para los componentes posteriores. Ese experimento se ejecuta 3 veces y tarda 5 minutos en terminarse cada vez. La pérdida de paquetes aumenta del 10 al 20 por ciento al 30 por ciento con cada ejecución, y el experimento tarda 15 minutos en total en completarse. Si tiene 100 despliegues paralelos, tendrá que esperar 1500 minutos para que se complete un solo experimento. Si tiene que ejecutar 10 experimentos, el impacto para sus desarrolladores sería insoportable. A gran escala, la ingeniería del caos necesita su propia canalización que le permita ejecutar experimentos en paralelo al proceso del ciclo de vida de desarrollo de software (SDLC).
- Implementaciones en Canary: las Islas Canarias proporcionan un entorno de pruebas para realizar experimentos de caos. Al dirigir un pequeño porcentaje del tráfico a un servicio de Canary o al utilizar métodos como la duplicación o la reproducción del tráfico, puede verificar los nuevos cambios en la infraestructura o el código sin que ello afecte a la estabilidad de su sistema de producción. Puede realizar experimentos a prueba de errores e introducir errores según sea necesario, ya que puede limitar el alcance del impacto en el usuario final.
- Experimentos programados: puede programar experimentos para verificar los mecanismos de recuperación predecibles de su aplicación. Utilice experimentos programados para reproducir eventos conocidos con frecuencia y ver cómo sus sistemas pueden recuperarse de eventos como la finalización de una instancia de EC2 detrás de un grupo de escalado automático, la eliminación de un pod de Kubernetes o la eliminación de una tarea de Amazon ECS.

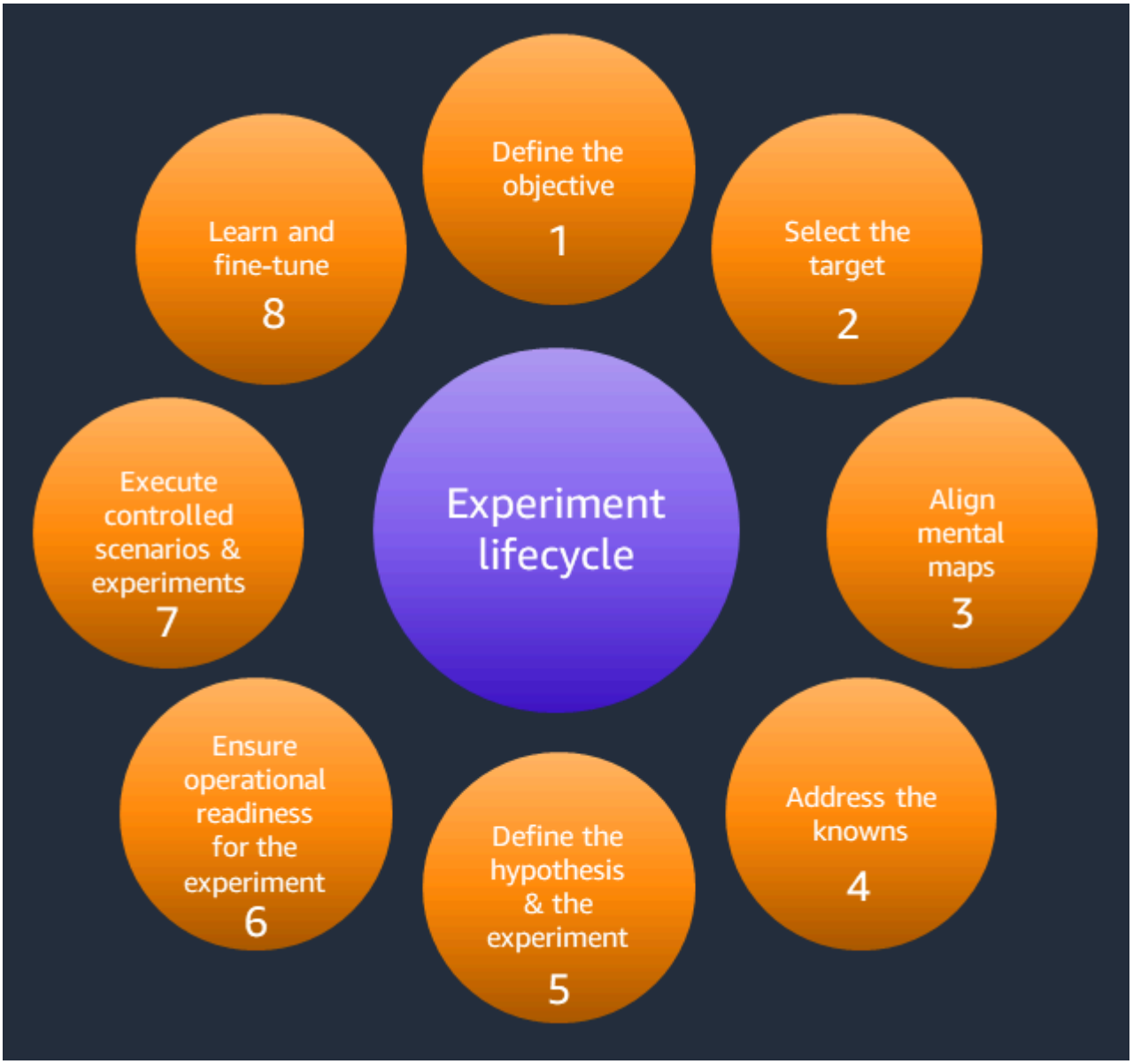
# Ciclo de vida continuo del experimento de ingeniería del caos

Como se ha explicado en la sección anterior, los experimentos de ingeniería del caos se pueden implementar de diferentes maneras. En todos los casos, la clave para crear un experimento de caos significativo es entender la aplicación, los incidentes históricos y las medidas correctivas implementadas, así como comprender claramente las áreas que se deben investigar, como la resiliencia o la seguridad. Sus conocimientos sobre la aplicación le ayudarán a formular una hipótesis sobre los posibles puntos débiles de la aplicación y a comprender cómo detectará, solucionará y recuperará una vez que se produzca el fallo.

El ciclo de vida del experimento del caos incluye los siguientes pasos:

1. Defina el objetivo del experimento.
2. Seleccione la aplicación de destino.
3. Alinee los mapas mentales.
4. Aborda los problemas conocidos de tu aplicación.
5. Defina la hipótesis y el experimento.
6. Garantice la preparación operativa para el experimento.
7. Ejecute escenarios y experimentos controlados.
8. Aprenda del experimento y ajústelo con precisión.

Estos pasos se ilustran en el diagrama y se analizan en las siguientes secciones.




# Defina los objetivos y establezca las expectativas

Antes de cada experimento, asegúrate de que tus objetivos y expectativas sean específicos, medibles, alcanzables, relevantes y estén sujetos a un plazo determinado. Defina claramente lo siguiente:

- Identifique las posibles fallas o debilidades en los sistemas y servicios para comprender cómo pueden afectar a los usuarios. Esto incluye identificar los posibles modos de fallo, como caídas del servidor, fallos de red o errores de software, y evaluar su posible impacto en el rendimiento y la fiabilidad generales del sistema.

- Cuantifique el impacto de las fallas definiendo los indicadores clave de riesgo (KRI) en sus sistemas y servicios. Esto incluye medir el efecto de las fallas cuando métricas como la latencia, el rendimiento y las tasas de error se desvían de su estado estable. Al comprender el impacto de dichas desviaciones, puede priorizar los esfuerzos para mitigar los fallos en función de los riesgos empresariales.
- Desarrolle y verifique estrategias para mitigar o prevenir las fallas. Esto incluye identificar posibles soluciones, como la redundancia, la corrección de errores o las estrategias alternativas, y probar su eficacia en un entorno controlado. Al verificar estas estrategias, puede asegurarse de que es eficaz a la hora de prevenir o mitigar los fallos y puede implementarlas en sus sistemas de producción con confianza.
- Mejore los procesos de respuesta a incidentes y recuperación ante desastres. Al reproducir los errores en un entorno controlado, puede probar los procesos de respuesta a incidentes, identificar posibles cuellos de botella o brechas y perfeccionar los procedimientos de recuperación ante desastres. Esto ayuda a garantizar que esté preparado para responder de forma rápida y eficaz en caso de que se produzcan fallos inesperados.

## Seleccione la aplicación de destino

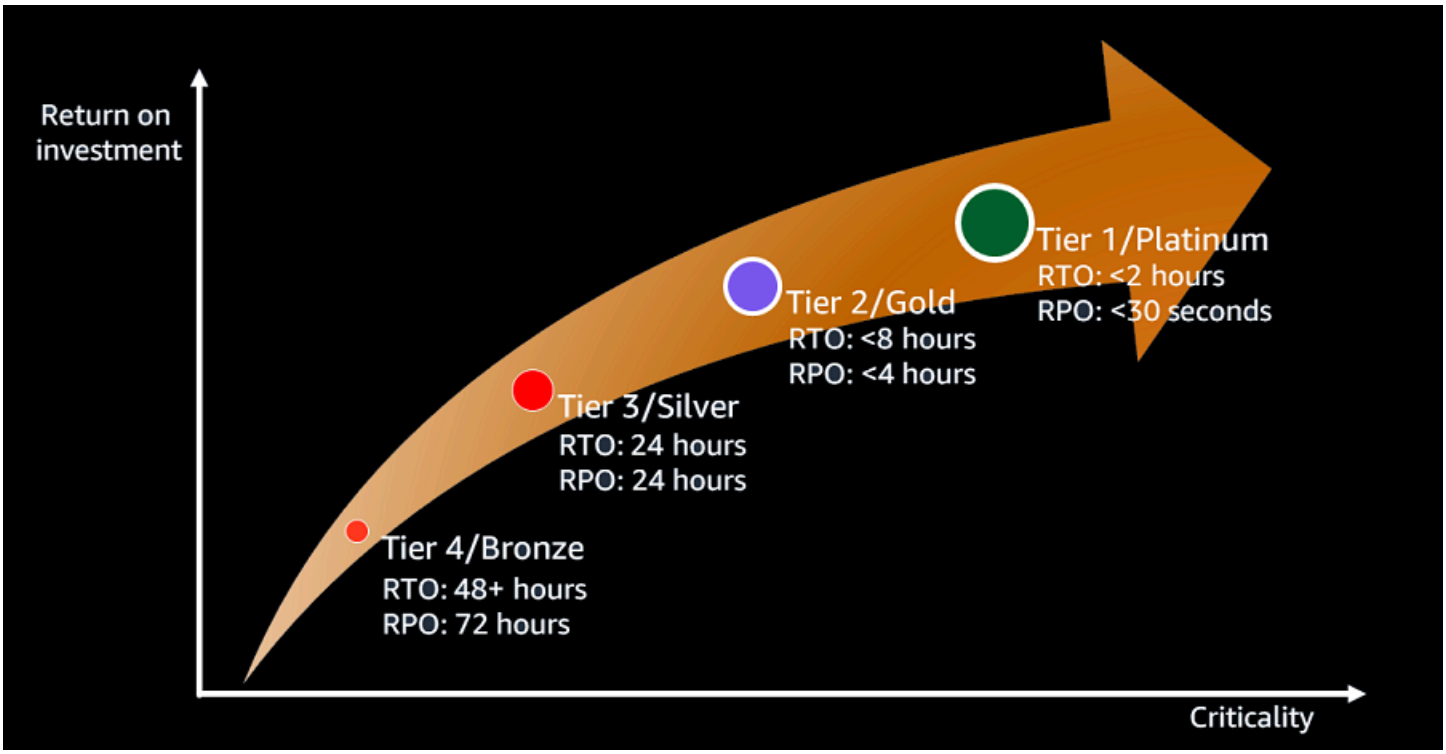
La ingeniería del caos es una técnica poderosa, pero requiere una priorización cuidadosa para maximizar el valor. A la hora de decidir dónde centrar los esfuerzos de ingeniería del caos, comience por considerar los servicios fundamentales de su empresa. Pida a sus equipos que repasen las etapas del ciclo de vida del desarrollo del software y comiencen primero a detectar errores en los entornos de prueba. Business-critical las aplicaciones están directamente relacionadas con los ingresos, la experiencia del cliente y las operaciones principales. Los experimentos de caos en estos servicios pueden descubrir vulnerabilidades que, si no se abordan, pueden afectar gravemente a la organización (y, potencialmente, a mercados enteros). Por ejemplo, céntrese primero en los servicios orientados al cliente, como los sistemas de negociación o los sistemas de pedidos. La priorización de estos servicios centrales proporciona la mayor protección por inversión de tiempo.

Después de los servicios esenciales, analice los componentes fundamentales, como las bases de datos, las colas de mensajes, las redes y las API de servicios compartidos. Es posible que se utilicen como componentes o servicios compartidos en toda la organización, por lo que su fallo provocará problemas generalizados. Confirmar la resiliencia de los servicios de infraestructura proporciona la confianza de que no paralizarán las aplicaciones dependientes que están por encima de ellos. Por ejemplo, un experimento de ingeniería del caos en el que se desactiva un clúster de Kafka revela muchos aspectos de la tolerancia a los fallos de las aplicaciones posteriores. Si bien

la infraestructura del sistema no está orientada directamente al cliente, es uno de los principales objetivos de la ingeniería del caos.

No olvide trazar un mapa de las brechas mentales de las personas, los procesos, la información de las instalaciones y las dependencias con terceros, ya que estas pueden provocar importantes perturbaciones si no se ajustan a los objetivos de tolerancia al impacto de su organización. Para obtener más información sobre cómo medir el ROI de la ingeniería del caos, lea Cuantificar el ROI de la ingeniería del caos en el documento de estrategia Invertir en la ingeniería del caos como una necesidad estratégica.

En el siguiente diagrama se muestra el retorno de la inversión que se obtiene al realizar experimentos de caos en diferentes niveles de servicios.




## Alinee los mapas mentales (descubrimiento de aplicaciones)

Cuando realices experimentos ad hoc o juegues, comenzarás el proceso de descubrimiento de aplicaciones con una sesión de pizarra que se centrará en mapear los detalles de tu aplicación. (Si llevas a cabo los experimentos de forma caótica, ya habrás alineado ese mapa mental al definir la aplicación de destino). Un buen método para entender las carencias mentales consiste en hacer que el miembro más joven del equipo dibuje primero un diagrama de la solicitud y, a continuación,

pida a otros miembros del personal de más alto nivel que lo añadan progresivamente. Esto revelará cualquier laguna de comprensión en los distintos niveles de experiencia.

El diagrama debe mostrar las dependencias directas de la aplicación, tanto en sentido ascendente como descendente, así como cualquier integración esencial de terceros. Asegúrese de que el flujo esperado de una solicitud a través de la aplicación esté alineado. Planifique los flujos de trabajo clave y los recorridos de los usuarios para obtener claridad sobre la forma en que los clientes utilizan la aplicación. Considere la posibilidad de utilizar un diagrama de secuencia para capturar esta información.

Tras esta sesión de colaboración, el equipo debería tener un modelo mental compartido de la aplicación, sus dependencias críticas y sus capacidades de supervisión, y comprender los riesgos para poder tomar la decisión fundamentada de continuar con un posible experimento de caos o cancelarlo.

## Resuelva los problemas conocidos de su aplicación

Los experimentos de ingeniería del caos están diseñados para detectar de forma proactiva los defectos de una aplicación. Al introducir errores como el aumento de la latencia, los reinicios del servidor o los cortes de energía en las zonas de disponibilidad, puede comprobar la capacidad de su aplicación para tolerar interrupciones realistas. Sin embargo, este proceso supone un nivel subyacente de estabilidad y estado en la aplicación de destino. Al ejecutar experimentos de caos en una aplicación que ya es problemática, se corre el riesgo de enmascarar problemas más profundos.

Antes de emprender una ingeniería de caos, los equipos deben resolver cualquier defecto, error o problema de rendimiento conocido en su aplicación.

## Defina la hipótesis y el experimento

Los incidentes pasados que causaron interrupciones en su aplicación o en otras aplicaciones de su organización pueden ser excelentes fuentes de ideas para hacer experimentos caóticos. Por ejemplo, ¿las interrupciones anteriores se debieron a errores de configuración o a la falta de patrones de resiliencia? Revisar los historiales de incidentes y reproducir las causas fundamentales de esos fracasos del mundo real mediante experimentos de caos es una forma eficaz de desarrollar la resiliencia ante problemas similares en el futuro.

Otra fuente valiosa de conceptos experimentales puede provenir directamente de los ingenieros, arquitectos y operadores que estén más familiarizados con una aplicación. Permitir que los

miembros del equipo presenten escenarios hipotéticos de fallo que, a su juicio, podrían interrumpir considerablemente la aplicación permite recopilar ideas basadas en información privilegiada. Luego, el equipo de aplicación puede evaluar cuál de estos escenarios propuestos podría tener el mayor impacto potencial o exponer los mayores riesgos desconocidos. Centrarse en experimentos de caos para escenarios de alto riesgo y menos entendidos puede generar importantes aprendizajes y prevenir problemas en el futuro.

Una tercera fuente de ideas proviene de la elaboración de modelos de resiliencia para anticipar las condiciones que provocarían pérdidas empresariales identificadas. Algunos ejercicios de modelado de resiliencia tienen un enfoque basado en componentes para construir un modelo de resiliencia, mientras que otros tienen un enfoque basado en sistemas. Un enfoque basado en componentes plantea la siguiente pregunta: «¿Qué sucede cuando el componente x está sometido a una carga extrema o ha fallado?» El equipo que desarrolla el modelo de resiliencia luego especula sobre el efecto de tal escenario en la aplicación más amplia e identifica los controles de monitoreo y prevención actualmente en vigor para detectar y mitigar los efectos del escenario. Como alternativa, un enfoque basado en sistemas sigue un proceso descendente para resaltar un estado no deseado de la aplicación (por ejemplo, «La tienda web muestra niveles de inventario obsoletos») e invita al equipo de la aplicación a anticipar qué condición o condiciones podrían provocar que la aplicación se comporte de esta manera.

## Asegúrese de que el experimento esté preparado desde el punto de vista operativo

Se necesitan indicadores cuantificables para medir el impacto de las condiciones adversas en la aplicación y su comportamiento, tal como se describió anteriormente en la sección sobre observabilidad. La capacidad de medir el comportamiento de la aplicación le permite determinar si las condiciones adversas afectaron a la aplicación y en qué magnitud.

La mejor manera de saber si hay un impacto en su aplicación es medir su estado estable. El estado estable mide el aspecto del funcionamiento normal y, por lo general, se alinea con los indicadores de experiencia empresarial y del cliente de una aplicación determinada. Antes de pasar al siguiente paso, asegúrate de disponer de la capacidad de observación necesaria para entender el impacto y de tener preparados los mecanismos de retroceso en caso de que el experimento no dé los resultados esperados.

# Realiza experimentos y escenarios controlados

En AWS, no recomendamos realizar un experimento de caos inicial en una aplicación que esté en producción. El objetivo de un experimento de caos es aprender algo nuevo sobre el comportamiento de la aplicación en condiciones de estrés. El comportamiento de la aplicación puede ser impredecible durante el experimento, por lo que realizar un experimento por primera vez en producción podría tener consecuencias para los clientes. Por lo tanto, siempre debes realizar un experimento de caos inicial en un entorno de nivel inferior que tenga un potencial mínimo de afectar a los usuarios del mundo real y, a continuación, repetir los entornos una vez que compruebes y estés seguro de que tu aplicación puede absorber las acciones introducidas, adaptarse y recuperarse de ellas.

Planifique cada experimento minuciosamente utilizando un documento que recoja los detalles clave, similar al documento de planificación del experimento que se incluye en el apéndice. Algunos de los campos críticos que se deben incluir son la definición del estado estacionario, la hipótesis y el método de inyección de fallas. La planificación, la ejecución y el análisis de un experimento de caos se pueden abarcar en un único artefacto.

Tras finalizar el plan escrito para el experimento, prepare el código necesario para introducir las interrupciones planificadas que se describen en el documento.

Para captar el impacto potencial durante el experimento, asegúrate de que existan mecanismos de observabilidad. Si aún no dispone de una forma automatizada de capturar los resultados del experimento, como los informes del AWS FIS experimento, identifique a los miembros del equipo que tomarán notas durante la ejecución, capture capturas de pantalla de los paneles y guíe al equipo durante el experimento.

# Aprenda y ajuste con precisión

Después de cada experimento, reúnanse en equipo para revisar y reflexionar sobre el experimento del caos. Haz un esfuerzo consciente por mantener una mentalidad irreprochable. Su objetivo debe ser tener un diálogo abierto y constructivo que se centre exclusivamente en obtener el máximo aprendizaje, no en culpar a los demás.

Comience por revisar la definición y la hipótesis del estado estacionario para el experimento. ¿Se comportó la aplicación como se esperaba? ¿Hubo alguna sorpresa que invalidara las suposiciones? Comenta las observaciones sobre cómo reaccionó la aplicación durante el experimento, tanto buenas como malas. Los datos recopilados (métricas, registros, capturas de pantalla, etc.) deberían explicar exactamente lo que ocurrió.

Aborde esta revisión de datos con curiosidad y no con criterio, e identifique las áreas en las que se pueden mejorar el diseño, la documentación, la supervisión u otras capacidades de las aplicaciones en función de lo aprendido. Estos elementos de acción se capturan como proyectos de seguimiento para hacer que la aplicación sea más resiliente.

Gracias a este enfoque irreprochable, puede mantener conversaciones sinceras sobre lo que salió mal y cómo solucionarlo. Suponga que todas las personas involucradas tienen intenciones positivas y confíe en que estaban trabajando para lograr buenos resultados. Su objetivo común es el crecimiento y el progreso de la organización a través del aprendizaje y la adaptación continuos. Las revisiones de los experimentos del caos, que se llevan a cabo de manera constructiva e impecable, proporcionan un espacio seguro para que su equipo obtenga información valiosa que haga que sus aplicaciones y su organización sean más confiables y resilientes a largo plazo. La atención se centra en los aprendizajes, no en las personas. Para difundir lo aprendido entre sus equipos, publique el informe de resultados del experimento en un lugar central y publique los resultados para que otros puedan aprender de él.

# Ampliar la ingeniería del caos en toda su organización

A medida que su organización adopte la ingeniería del caos, estandarizarla e implementarla presentará desafíos. En las primeras etapas de madurez, es probable que los diferentes equipos utilicen diferentes herramientas y variaciones del proceso de ingeniería del caos descrito en las secciones anteriores. Al mismo tiempo, es posible que algunos equipos no prioricen o adopten en absoluto la ingeniería del caos, a pesar de sus posibles beneficios. Las siguientes secciones proporcionan orientación sobre cómo superar estos desafíos.

En general, su enfoque de la ingeniería del caos debe estar diseñado para lograr un equilibrio entre el liderazgo centralizado y la participación descentralizada. Este equilibrio ayuda a garantizar que la ingeniería del caos se integre en el proceso de desarrollo y que los aprendizajes se compartan en toda la organización.

## Establecer una práctica de ingeniería del caos

La estandarización de la práctica de la ingeniería del caos puede acelerar su adopción. Compartir los aprendizajes de los experimentos entre los equipos puede aumentar el rendimiento de las inversiones en ingeniería del caos.

Construye un centro de excelencia centralizado o reúne a un grupo de expertos en la materia como parte de tu práctica de ingeniería del caos. Al tratarse de una función pequeña y centralizada, este equipo puede trabajar en los equipos de desarrollo de software, infraestructura, seguridad y negocios, y mantener los estándares que utilizan esos equipos. Para simplificar, el centro de excelencia se denomina equipo de práctica centralizada y, en el resto de esta guía, los grupos que aplican la ingeniería del caos se denominan equipos de práctica.

### Función del equipo de práctica centralizada

El equipo de práctica centralizada es responsable de desarrollar e implementar prácticas de ingeniería del caos en toda la organización. Trabajan en estrecha colaboración con los equipos en ejercicio para guiarlos en el diseño y la realización de experimentos y garantizar que los experimentos sean valiosos para la empresa. El equipo de práctica centralizada también proporciona orientación y apoyo a los equipos de desarrollo, infraestructura y seguridad para ayudarlos a integrar la ingeniería del caos en sus procesos de desarrollo.

Las principales responsabilidades de un equipo de práctica de ingeniería del caos centralizado incluyen las siguientes:

- Capacitación: una función de ingeniería del caos centralizada facilita la introducción de la práctica de la ingeniería del caos a través de jornadas de juego y talleres. Guían a los equipos en el proceso de creación del caos, lo que incluye la selección de escenarios de fallo, la definición de hipótesis y la elaboración de informes para compartirlos con el resto de la organización. El equipo de práctica centralizado debe ser propietario de los materiales de formación y trabajar para mejorar las aptitudes de los equipos de práctica en el uso de la ingeniería del caos.
- Asesoramiento: el equipo de práctica centralizada también puede actuar como asesor para supervisar los experimentos que llevan a cabo los equipos de práctica. Su experiencia y conocimientos pueden garantizar que los experimentos aporten valor a la empresa y se lleven a cabo de forma segura. Del mismo modo, el equipo puede supervisar la ejecución y el resumen de un experimento para guiar a las personas que se inician en la ingeniería del caos.
- Marketing y seguimiento del valor: comunicar el valor empresarial de la ingeniería del caos es clave para el éxito de un programa de este tipo. Cada equipo que participe en los experimentos de ingeniería del caos debe recopilar datos de los experimentos realizados en toda la empresa y demostrar el valor de la inversión de la organización en la ingeniería del caos. Esto incluye cuantificar y celebrar el número de incidentes que se evitaron durante cada experimento, el tiempo de inactividad en el que se habría incurrido si el experimento hubiera fracasado y el impacto general en la empresa si los escenarios de fallo se hubieran producido en la producción. Al recopilar y centralizar estos datos de todos los equipos y ponerlos a disposición de toda la organización, el equipo de práctica centralizada puede hacer un seguimiento del valor derivado de la adopción de la ingeniería del caos en toda la organización e influir en él.
- Normas: el equipo de práctica centralizado debe ser el propietario y el mantenimiento del proceso de realización de experimentos de caos, de las plantillas de planificación y elaboración de informes sobre los experimentos y de las herramientas utilizadas para llevarlos a cabo.

  El equipo central debe poseer y gestionar las plantillas de planificación de experimentos, las plantillas de informes de experimentos, la documentación de los procesos y los materiales de capacitación. La documentación sobre las mejores prácticas y los materiales de capacitación proporcionan orientación a los equipos en prácticas sobre temas como las barandillas que pueden utilizar para limitar el impacto de un experimento, cuándo realizar un experimento en producción y cómo evolucionar su uso de la ingeniería del caos a lo largo del tiempo. Para ver ejemplos de plantillas y resultados, consulte el apéndice.

  El equipo de práctica centralizada también debe ser el propietario del proceso de realización de un experimento, incluidas las comunicaciones y la escalación, y de cuándo y cómo comunicarse con

otros equipos de la organización antes o durante el experimento. El proceso también debe indicar cuándo se requieren barandas.

El equipo de práctica centralizada también debe seleccionar y poseer las herramientas básicas para realizar experimentos de caos (por ejemplo, herramientas como AWS FIS estas). La selección e implementación de herramientas complementarias, como las herramientas de generación de carga, deben ser decididas por los equipos de práctica. Los equipos en prácticas deberían poder adaptar el proceso general y las herramientas para que se adapten mejor a sus necesidades.

### El papel de los equipos practicantes

El equipo centralizado es responsable de impulsar la estrategia general de ingeniería del caos, mientras que los equipos que practican participan en el proceso y son los responsables del desarrollo y la ejecución de los experimentos. Esto ayuda a garantizar que los experimentos sean relevantes para cada producto o servicio específico, y que los aprendizajes sean procesables y puedan aplicarse para mejorar la confiabilidad y la resiliencia del producto. El equipo de práctica centralizada actúa como mentor y responsable de los estándares y procesos de ingeniería del caos de la organización. Sin embargo, para evitar que el equipo centralizado se convierta en un obstáculo, los equipos de práctica individuales deberán aprender del consultorio central y realizar por sí mismos experimentos sobre el caos.

## Establecer una comunidad de práctica

Además de crear un equipo centralizado, le recomendamos que establezca una comunidad informal de profesionales interesados en la ingeniería del caos. Esta comunidad proporciona una plataforma para compartir conocimientos, mejores prácticas y experiencias entre los equipos en ejercicio y la organización en general.

La comunidad de práctica puede ser gestionada por un equipo centralizado de práctica de ingeniería del caos, pero cualquier persona de la organización puede convertirse en miembro de la comunidad. El equipo centralizado puede aprovechar la comunidad de práctica para difundir actualizaciones y obtener información sobre lo aprendido, así como para recopilar comentarios de los equipos en ejercicio que utilizan los estándares y los procesos gestionados por el equipo centralizado. La comunidad actuará como un circuito de retroalimentación para informar al equipo centralizado sobre la eficacia de las prácticas de ingeniería del caos en todos los equipos en prácticas. Luego, el equipo

de práctica centralizada podrá ajustar su documentación y los elementos de apoyo para brindar un mejor apoyo a los equipos de productos.

## Incorporar la ingeniería del caos a su resiliencia operativa

Un experimento de caos es una inversión de su empresa para evitar incidentes en la producción. Será necesario determinar dónde la empresa puede obtener el mayor rendimiento de esta inversión. La organización puede trabajar con el equipo de práctica de ingeniería del caos centralizado para actualizar sus estándares y determinar qué productos son lo suficientemente críticos como para requerir la experimentación del caos.

Proceso de desarrollo de sistemas

La ingeniería del caos y los experimentos del caos deben realizarse repetidamente como parte del ciclo de vida de una aplicación. De forma similar a como los equipos realizan habitualmente las pruebas de recuperación ante desastres, deberían llevar a cabo experimentos de caos y jornadas de juego de forma continua y periódica durante todo el año. Este enfoque mejora la forma en que una organización anticipa, observa y responde a los incidentes.

# Conclusión

La disciplina de la ingeniería del caos ha avanzado mucho en la última década. Se ha adoptado en diversos sectores y ha ayudado a las organizaciones a crear servicios resilientes y a aumentar la satisfacción de los clientes. (Para ver ejemplos de cómo las organizaciones han implementado estas prácticas, consulte Chaos Engineering Stories). La ingeniería del caos permite a las organizaciones reducir los riesgos de sus aplicaciones de misión crítica al introducir errores controlados en todos los niveles de su conjunto de aplicaciones, incluidos los servicios de los proveedores de servicios en la nube. Tener la capacidad de afectar a todo un conjunto de aplicaciones de forma controlada permite una resiliencia continua, una mejora de la excelencia operativa, la observabilidad y una arquitectura orientada a la recuperación sin el stress de las interrupciones de producción. Este enfoque también conduce a mejores prácticas de pruebas de resiliencia en toda la organización. Para empezar a adoptar la ingeniería del caos, organice un día o un taller sobre el caos para demostrar el valor que la ingeniería del caos puede aportar a su organización.

# Recursos

AWS FIS implementaciones de modelos de fallas:

- Úselo AWS Fault Injection Service para demostrar la resiliencia de las aplicaciones en múltiples regiones y zonas de disponibilidad (AWS entrada del blog)
- AWS El simulador de inyección de fallas apoya los experimentos de ingeniería del caos en los Amazon EKS Pods (entrada AWS del blog)
- Anunciamos las nuevas funciones de AWS Fault Injection Simulator para las cargas de trabajo de Amazon ECS (AWS entrada del blog)
- Mejore la resiliencia de las aplicaciones con las métricas de volumen de Amazon EBS y el simulador de inyección de AWS fallos (AWS entrada del blog)
- La ingeniería del caos aprovecha el simulador de inyección de AWS fallos en un AWS entorno con varias cuentas (entrada del blog)AWS
- Experimentos de caos en Amazon RDS con AWS Fault Injection Simulator (AWS entrada del blog)
- Creación de aplicaciones resilientes sin servidor mediante la ingeniería del caos (AWS entrada del blog)
- Utilice FIS para interrumpir una instancia puntual (taller)AWS
- Cómo automatizar la ingeniería del caos en sus procesos de entrega (publicación de la comunidad)AWS

Otros recursos:

- Invertir en la ingeniería del caos como necesidad estratégica
- Historias de ingeniería del caos
- CloudWatchDocumentación de Amazon
- Documentación de Amazon CloudWatch RUM
- Documentación de AWS X-Ray
- Documentación de AWS FIS

# Apéndice: Documentos de muestra

Los documentos de muestra que se proporcionan en esta sección utilizan un sitio ficticio de adopción de mascotas (PetSite) como aplicación objetivo para un experimento de caos.

- Documento de planificación del experimento
- Documento de resultados del experimento

## Documento de planificación del experimento

### Estado estacionario

| Process name (Nombre del proceso) | Un sitio de adopción de mascotas |
|---|---|
| Arquitectura física | (Enlace al diagrama de arquitectura.) |
| Arquitectura lógica | (Enlace al diagrama lógico). |
| Defina el estado estacionario | El tiempo medio de carga de la página web de adopción de mascotas, medido con Largest Contentful Paint (LCP), es de 2,5 segundos o menos, con una latencia del 99 por ciento (P99) de 4 segundos o menos, con una base de referencia de 5000 usuarios simultáneos. |
| Métricas de estado estacionario | Métrica de LCP capturada en toda la base de usuarios y métricas doradas (latencia, rendimiento, tasas de error, saturación). |
| Observabilidad en estado estacionario | El navegador del usuario capturará el LCP, lo enviará a Amazon CloudWatch y lo inspeccionará con CloudWatch RUM. Durante un período de 60 segundos, se sumarán la media y el tiempo de LCP de 99 pesos para todas las solicitudes de ese período. Top-level Las |

| Process name (Nombre del proceso) | Un sitio de adopción de mascotas |
| --- | --- |
| | métricas de oro se capturan mediante el uso de. CloudWatch |
| Proceso para alcanzar el estado estable | Grafana K6 se utilizará para crear una carga que simule los niveles normales de tráfico de producción de aproximadamente 5000 usuarios simultáneos. |

## Requisitos de observabilidad

Los equipos deberían poder ver lo siguiente:

- Estado estacionario: ¿Qué se observará para verificar que la aplicación esté en condiciones normales?
- Condición de fallo: ¿Cómo aparecerá el estado de fallo en el salpicadero? Por ejemplo:
  - Alarmas que deberían activarse
  - Registros que deben generarse
- Impacto de la falla: ¿Qué se debe observar para ver los componentes que se espera que se vean afectados (alcance del impacto)?
- Recuperación: ¿Cómo se visualizará y medirá la recuperación para captar el MTTR?
- Depuración: detalles de solución de problemas relacionados con los errores de los experimentos.

La siguiente tabla proporciona sugerencias y ejemplos para un gráfico de requisitos de observabilidad. Debe definir lo que debe observarse en función de su experimento específico.

| ¿Qué hay que observar | Enlace a la herramienta de observabilidad | ¿Qué se está observando |
| --- | --- | --- |
| Fuente de entrada | Cuadro de mandos Grafana K6 | • Recuento de contenedores en ejecución<br>• Solicitudes por segundo |

| ¿Qué hay que observar | Enlace a la herramienta de observabilidad | ¿Qué se está observando |
|---|---|---|
| Estado general de la aplicación | • CloudWatch Panel de adopción de mascotas<br>• Panel de experiencia de usuario (RUM) en materia de adopción de mascotas | • Recuento de nodos en buen estado de Amazon EKS<br>• Utilización de la CPU del nodo Amazon EKS |
| Estado del flujo de trabajo | CloudWatch Panel de adopción de mascotas | Tiempo de LCP, métricas doradas |
| Rastros | Panel de adopción de X-Ray mascotas | • Solicita latencia<br>• Número de solicitudes<br>• Recuento de errores |
| Registros | CloudWatch Registros de adopción de mascotas | Todos los errores que detecten los pods se enviarán a CloudWatch Logs. |

## Definición de experimento

| Nombre del experimento | Amazon EKS PetSite pod CPU stress |
|---|---|
| Experimenta con el código fuente | (Enlace al repositorio de fuentes del experimento). |
| Descripción del experimento | Este experimento explora cómo un aumento en el uso de la CPU del módulo de PetSite aplicaciones afectaría a la experiencia general del cliente. Al inyectar stress de CPU en cada PetSite módulo en ejecución, podremos |

| Nombre del experimento | Amazon EKS PetSite pod CPU stress |
| --- | --- |
| | entender si hay un impacto en los clientes y el alcance de ese impacto. |
| Requisitos o parámetros del experimento | Carga de aplicación: media de producción<br><br>Selector de etiquetas para cápsulas: `app=petsite` |
| Duración del experimento | 10 minutos |
| Entorno | Entorno de prueba alfa |
| Experimenta con los recursos objetivo | PetSite pods de aplicaciones |
| Experimente la línea base que se introduce mediante la herramienta de generación de carga | • El 54% de las solicitudes tienen un LCP inferior a 2,5 segundos.<br>• El 46% de las solicitudes tienen un LCP inferior a 4 segundos.<br>• No se observa ningún error. |
| Condición de retroceso | Ninguno |

## Hipótesis

| ¿Qué pasaría si | Impact | Recuperación |
| --- | --- | --- |
| ¿Qué pasaría con el estado estable si los pods de PetSite aplicaciones utilizaran o provocaran un uso de la CPU superior al 60% durante 10 minutos con un tráfico normal a nivel de producción? | Los tiempos de LCP se mantendrán por debajo de 2,5 segundos para un P50 de los usuarios con un P99 de 4 segundos o menos. El consumidor debería poder cargar la página de destino. PetSite | Detección:<br>• Las alarmas configuradas en detectarán el stress de la CPU CloudWatch.<br>• Las métricas del LCP también generarán alarmas |

| ¿Qué pasaría si | Impact | Recuperación |
| --- | --- | --- |
| | | ante la degradación de la experiencia del usuario.<br><br>Self-healing:<br><br>• La naturaleza distribuida de la arquitectura de microservicios significa que muchas instancias de pods se ejecutan en varias zonas de disponibilidad.<br><br>• El plano de control del clúster de EKS desviará el tráfico de los módulos afectados y lanzará nuevos módulos en los nodos de trabajo.<br><br>Recuperación:<br><br>Cuando el uso de la CPU vuelva a la normalidad, el LCP debería recuperarse automáticamente. |

## Proceso de experimentación

Adapte el siguiente proceso paso a paso de ejemplo a su experimento específico:

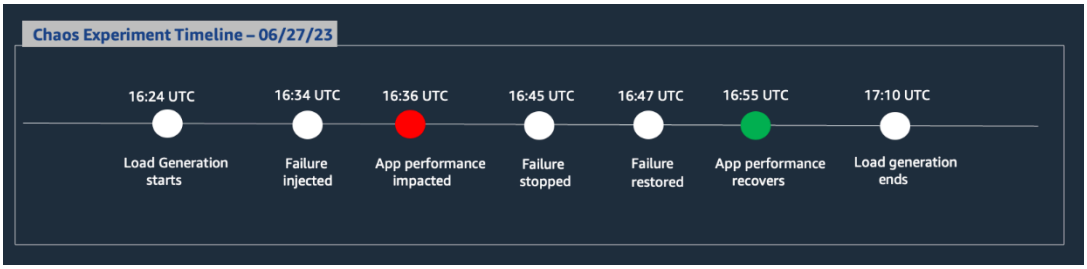
1. Valide el acceso y la funcionalidad de todos los Amazon CloudWatch, CloudWatch RUM y AWS X-Ray paneles de control.
2. Valide el estado del entorno de la aplicación:

a. Confirme que el clúster de EKS está en buen estado mediante el CloudWatch panel de control.
b. Visite la implementación de la aplicación del sitio de prueba de adopción de mascotas en la URL de ejemplo.

3. Inicie una carga para alcanzar un estado estable:
   a. Confirme que el generador de carga esté funcionando y enviando 5000 solicitudes por segundo.
   b. Espere 5 minutos para que la aplicación alcance su estado estable.
   c. Confirme el estado estable de la aplicación mediante el panel de control de CloudWatch RUM.
4. Iniciar una avería (experimento):
   a. Abre la AWS FIS consola.
   b. Realiza el experimento de adopción de mascotas, cápsula de estrés.
   c. Confirma que el experimento se está ejecutando en la consola.
5. Observe el impacto de la falla en su aplicación:
   a. Realice capturas de pantalla del CloudWatch RUM y de los CloudWatch paneles de control y anote cualquier dato anómalo.
   b. Una vez finalizado el experimento AWS FIS, captura capturas de pantalla adicionales para registrar si la aplicación vuelve al estado estable en ausencia de stress y observa cualquier anomalía en los puntos de datos.
   c. Si el estado estable no se reanuda, tome medidas para recuperar la aplicación y registre las medidas adoptadas.
6. Valide que el entorno haya vuelto a la normalidad:
   - Revise todas las métricas empresariales, de la experiencia del usuario, de las aplicaciones y de la infraestructura para comprobar que el sistema ha vuelto a un estado conocido. Haga capturas de pantalla del panel de control si resulta útil.

## Cronología del experimento

Asegúrese de capturar la cronología del experimento de principio a fin, empezando por la generación de la carga, la introducción de la falla, la observación del impacto y la recuperación de la aplicación, y finalizando cuando detenga la generación de la carga. Esto se ilustra con el siguiente ejemplo.




## Resultados de un experimento

| ID de ejecución del experimento | Resultados del experimento |
| --- | --- |
| `PET-ADOPT-EXP-23` | (Enlace a los resultados del experimento). |

## Defectos identificados

- El clúster de Kubernetes no detectó el deterioro de la CPU de los PetSite pods, por lo que no programó despliegues adicionales.
- Como resultado de este experimento, las tasas de error no aumentaron ni 4XX ni 5XX.
- Tenemos que ajustar la comprobación del estado del módulo para tener en cuenta el impacto en el LCP cuando hay limitaciones de recursos.

# Documento de resultados del experimento

## Configuración

Documente las configuraciones específicas del experimento. Por ejemplo:

- La generación de carga está configurada para simular que 5000 usuarios emiten un total de 85 solicitudes por segundo.

## Requisitos previos

- Se verificó que el sitio de adopción de mascotas funcionaba en el entorno de prueba alfa.

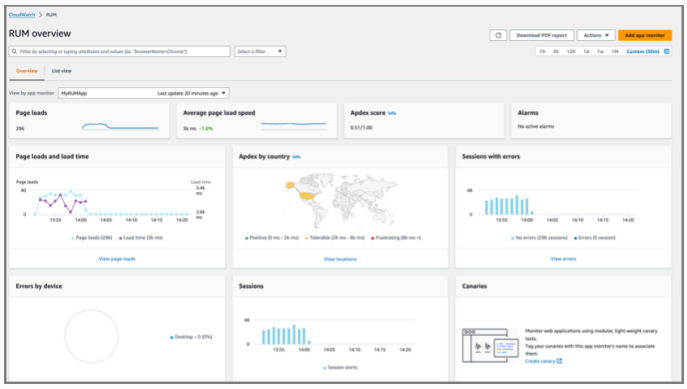
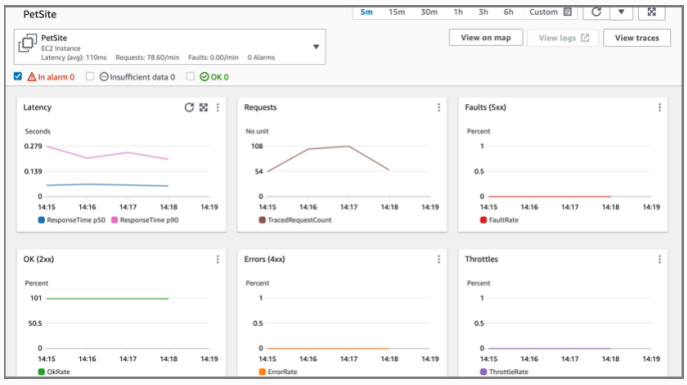
- Se verificó que la plantilla del experimento estaba configurada para aplicar stress de CPU a los pods de PetSite aplicaciones que se ejecutan en el clúster de EKS.  Los pods de aplicaciones se identificaban con la etiqueta Kubernetes. `app=petsite`
- Se confirmó que Load estaba en ejecución y generaba 85 solicitudes por segundo.

## Estado estacionario

Documente los pasos dados para alcanzar el estado estable y cómo lo verificó. Por ejemplo:

Para la implementación de prueba de un sitio de adopción de mascotas, se generará una carga de 85 RPS para simular el estado estable. Antes de la ejecución del experimento, se revisaron el CloudWatch RUM y los CloudWatch cuadros de mando para comprobar que todas las métricas empresariales y de las aplicaciones se encontraban dentro de los rangos normales.

Datos de observabilidad:

| Expected | Observado |
| --- | --- |
| • El LCP dura menos de 4 segundos para el 99% de las solicitudes.<br>• La latencia de respuesta es inferior a 500 ms.<br>• No hay errores 4XX ni 5XX. |  <br>  |

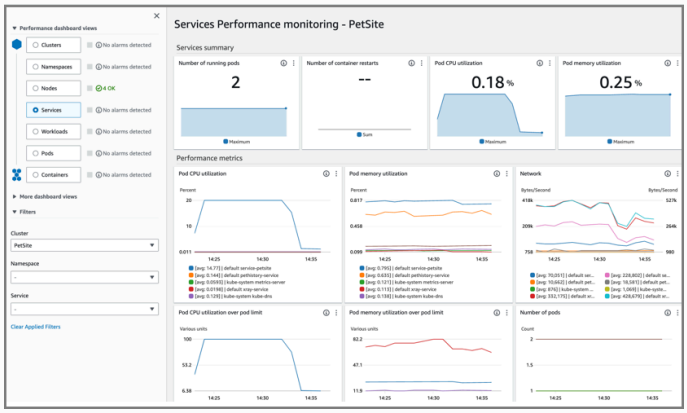
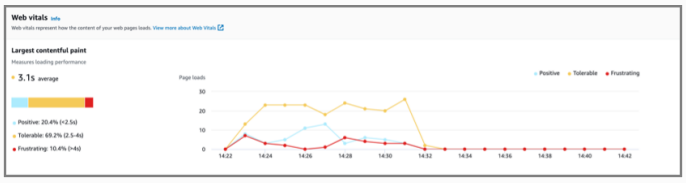
## Inyección de errores

AWS FIS se utilizó para detectar fallas utilizando la plantilla del experimento (proporcione el enlace). El experimento estaba programado para ejecutarse durante 10 minutos y se configuró una reversión si los nodos trabajadores experimentaban un estrés de CPU superior al 60 por ciento.

## Observación de fallos

Se revisaron el CloudWatch RUM y los CloudWatch paneles de control para realizar un seguimiento del estado estable de la aplicación (definido mediante métricas de LCP). Las capturas de pantalla se capturaron en la siguiente tabla.

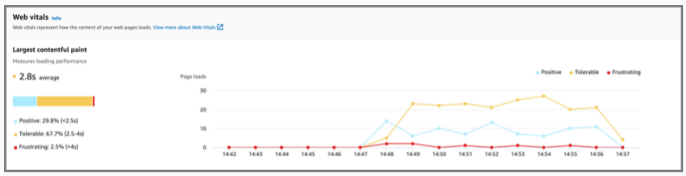
Datos de observabilidad:

| Expected | Observado |
| --- | --- |
| • El LCP debe permanecer por debajo de 4 segundos para el P99.<br>• El tiempo de respuesta debe permanecer por debajo de 500 ms.<br>• No se deben encontrar errores 4XX o 5XX. |  <br>  |

## Recuperación

Una vez eliminada la tensión (el AWS FIS experimento se ha completado y se ha eliminado la tensión de la CPU de los módulos), la aplicación debería volver a su estado estable normal. No debería ser necesaria ninguna intervención manual.

Datos de observabilidad:

| Expected | Observado (captura de pantalla) |
| --- | --- |
| El LCP P99 debe durar menos de 4 segundos con una media inferior a 2,5 segundos. |  |

# 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 | — | 4 de abril de 2025 |

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.