Erhöhung der Widerstandsfähigkeit und Verbesserung des Kundenerlebnisses durch den Einsatz von Chaos Engineering auf AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Erhöhung der Widerstandsfähigkeit und Verbesserung des Kundenerlebnisses durch den Einsatz von Chaos Engineering auf AWS

Table of Contents

Erhöhung der Widerstandsfähigkeit und Verbesserung des Kundenerlebnisses durch den Einsatz von Chaos Engineering auf AWS

Laurent Domb, Cheftechnologe, Federal Financials, Amazon Web Services

April 2025 (Geschichte des Dokuments)

Chaos Engineering ist die Disziplin, bei der mit einer Anwendung experimentiert wird, um Vertrauen in die Fähigkeit Ihres Unternehmens und Ihrer Anwendung aufzubauen, turbulenten Produktionsbedingungen standzuhalten. Dabei handelt es sich um einen proaktiven Ansatz zur Steigerung der Ausfallsicherheit mit dem Ziel, zu überprüfen, ob Ihre Anwendung und Ihr Unternehmen in der Lage sind, Servicebeeinträchtigungen zu absorbieren, sich daran anzupassen und sich letztendlich von ihnen zu erholen, indem kontrollierte Ausfälle bei Mitarbeitern, Prozessen und Technologien eingeführt werden. Ziel ist es auch, Schwachstellen zu identifizieren und zu beseitigen, bevor sie zu Ausfällen oder anderen Produktionsunterbrechungen führen können.

Wir bei Amazon wissen, dass Ausfälle in verteilten Systemen unvermeidlich sind, sodass das Funktionieren trotz vorhandener Fehler ein normaler Betriebsmodus ist. Da Interaktionen zwischen Diensten unweigerlich zum Scheitern verurteilt sind, müssen Sie verstehen, wie Ihre Services auf verschiedene Ausfallmodi reagieren, und Services entwickeln, die gegen wichtige Sicherheitslücken wie Abhängigkeitsausfälle, Wiederholungsstürme, beeinträchtigte Availability Zones und Erschöpfung der Hostressourcen widerstandsfähig sind.

Nehmen wir das Beispiel eines Storms mit Wiederholungsversuchen. Ein lokalisierter Fehler in einem Client kann sich erheblich auf mehrere Dienste auswirken. Dies wird allgemein als Schmetterlingseffekt bezeichnet. Ein Wiederholungssturm ist eine Ausprägung des Butterfly-Effekts, bei dem eine fehlgeschlagene Abhängigkeit die Clients und die Clients dieser Clients dazu veranlasst, den fehlgeschlagenen Vorgang erneut zu versuchen, was zu einem exponentiellen Anstieg des Datenverkehrs führt. Dienste werden überlastet, weil sie zusätzlich zu Wiederholungsversuchen auch auf regulären Datenverkehr reagieren und gleichzeitig Leistungseinbußen bewältigen müssen.

Chaos Engineering ist eine Reaktion auf die zunehmende Komplexität verteilter Systeme. Es handelt sich um einen multidisziplinären Ansatz, der Prinzipien der Chaostheorie, des Systemdenkens und der Technik kombiniert, um komplexe Systeme zu entwerfen und zu verwalten, die unerwarteten

Ereignissen und Verhaltensweisen standhalten. Im Kern befasst sich die Chaos-Technik mit dem Verständnis und der Steuerung des Verhaltens komplexer Systeme unter Bedingungen der Unsicherheit und Unvorhersehbarkeit. Es wird anerkannt, dass traditionelle technische Ansätze, die auf der Vorhersage und Kontrolle von Ergebnissen beruhen, oft nicht ausreichen, um mit der komplexen und dynamischen Natur verteilter Systeme umzugehen. Wenn diese Systeme immer größer werden, überschreiten sie oft den Kenntnisstand einer einzelnen Person.

Chaos Engineering stellt Konzepte, Techniken und Tools zur Verfügung, mit denen Systeme bewusst mit Fehlern behaftet werden können, um Schwachstellen aufzudecken, bevor sie sich in der Produktion bemerkbar machen. Durch diesen proaktiven Ansatz können Unternehmen darauf vertrauen, dass ihre Systeme auch unter stressigen Bedingungen funktionieren. Obwohl Chaos Engineering immer noch eine sich entwickelnde Praxis ist, stellt es einen grundlegenden Wandel hin zu der Entwicklung, Verwaltung und dem Betrieb moderner Computersysteme dar, die angesichts der zunehmenden Komplexität und Vernetzung widerstandsfähig sind.

In den folgenden Abschnitten dieses Leitfadens werden die Vorteile von Chaos Engineering erörtert, die Durchführung von Chaos-Engineering-Experimenten erläutert und die Ansätze beschrieben, mit denen Sie Chaos Engineering in großem Umfang in Ihrem Unternehmen implementieren können. Ebenfalls enthalten sind Musterdokumente zur Versuchsplanung und zu den Versuchsergebnissen, die Sie als Vorlagen für Ihre Experimente im Bereich Chaos Engineering verwenden können.

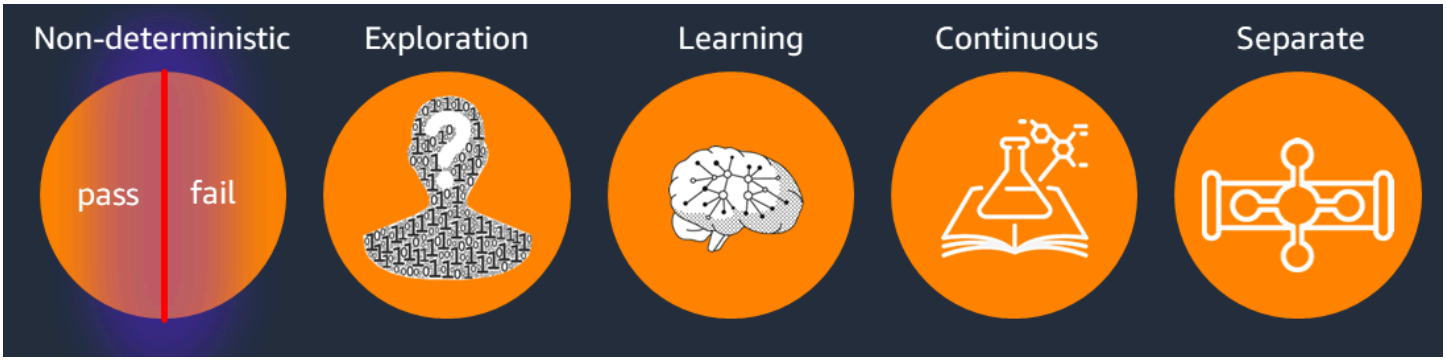
Im nächsten Abschnitt wird untersucht, wie sich die Eigenschaften von Chaos Engineering von herkömmlichen Resilienztests wie Unit-, Smoke- oder Integrationstests unterscheiden.

-Übersicht

Vergleich von Resilienztests mit Chaos-Engineering

Resilienztests sind deterministisch. Das heißt, sie validieren bekannte Merkmale von Resilienzmechanismen wie Schutzschaltern, Wiederholungsversuchen, Failovers oder Fallbacks, die in Ihrer Anwendung implementiert wurden. Es bestätigt, wie diese Anwendungskomponenten kontrollierte Störungen mit minimalen bis keinen Auswirkungen auf die Benutzer abfangen. Daher konzentrieren sich Resilienztests auf die Validierung bekannter Fehlerquellen, die in Anwendungskomponenten eingebracht werden, um pass/fail Ergebnisse zu erzielen. Sie sollten als einen Schritt in Ihrer Pipeline kontinuierlich Resilienztests durchführen, um sicherzustellen, dass Ihre Resilienzsituation nicht beeinträchtigt wird. Bei Resilienztests führen Sie häufig keine Tests mit echten Komponenten durch, sondern simulieren APIs, die eine bestimmte Komponente simulieren. Dieser Ansatz ermöglicht konsistente, reproduzierbare Tests von Ausfallszenarien in einer kontrollierten Umgebung und eignet sich daher für automatisierte Pipeline-Integrations- und Regressionstests.

Im Gegensatz dazu ist Chaos Engineering nicht deterministisch. Das heißt, es basiert auf Hypothesen und verifiziert Ihr mentales Modell darüber, wie die Anwendung und ihre Abhängigkeiten (Menschen, Prozesse und Technologie) unvorhergesehene Ausfälle absorbieren, sich daran anpassen und sich schließlich von ihnen erholen. Daher konzentriert sich Chaos Engineering auf die umfassende Überprüfung unbekannter Fehlerquellen mit dem Ziel, Fehler frühzeitig zu erkennen und zu beheben, bevor sie zu größeren Ereignissen werden. Chaos Engineering fördert kontinuierliches Lernen und sollte im Rahmen einer separaten Chaos-Pipeline oder durch Ad-hoc-Experimente geübt werden, die es Ihnen ermöglichen, mehrere Experimente zu einem beliebigen Zeitpunkt durchzuführen, ohne die Produktivität Ihres Entwicklers bei der Codebereitstellung zu beeinträchtigen.



Der Chaos-Engineering-Prozess beginnt häufig mit einem Chaos-Spieltag. Dabei handelt es sich um eine spezielle Veranstaltung, bei der Teams bewusst kontrollierte Fehler oder Ausfälle in ihre Anwendung einbauen. Der Spieltag ist progressiv: Er beginnt in Umgebungen auf niedrigerer Ebene (z. B. beim Entwickeln oder Testen) und wird mit steigendem Selbstvertrauen schrittweise in Umgebungen mit höherer Ebene (wie Staging und Vorproduktion) überführt. Durch die systematische Bearbeitung dieser Umgebungen können Teams überprüfen, ob ihre Systeme die eingeführten Fehler ordnungsgemäß tolerieren, bevor sie in die Produktion gehen. Dieser methodische Fortschritt stellt sicher, dass die Teams bis zu dem Zeitpunkt, zu dem Chaosexperimente in Produktionsumgebungen durchgeführt werden, großes Vertrauen in die Widerstandsfähigkeit ihres Systems aufgebaut haben. Der Game Day-Prozess ist ein proaktiver Ansatz zur Identifizierung von Schwächen und Schwachstellen in der Architektur und den betrieblichen Abläufen einer Anwendung und beseitigt gleichzeitig den Lernstress bei einem unerwarteten Produktionsausfall.

Der Wert von Chaos Engineering

Komplexe Systeme sind in der heutigen Welt allgegenwärtig. Sie spielen in vielen Bereichen unseres Lebens eine entscheidende Rolle, von den Finanzmärkten bis hin zum Gesundheitswesen. Wir erwarten, dass diese Systeme immer betriebsbereit sind. Komplexe Systeme sind jedoch häufig anfällig für unerwartete Ereignisse und Verhaltensweisen, die erhebliche Folgen haben können. Organizations müssen für Störungen planen, anstatt sich zu fragen, ob sie eintreten werden. Sie können dies tun, indem sie Szenariotests für ihre kritischen oder geschäftskritischen Unternehmensdienste anwenden. Hier kommt Chaos Engineering ins Spiel.

Chaos Engineering bietet einen Ansatz zur Verwaltung komplexer Systeme, der dazu beitragen kann, Risiken zu minimieren und die Widerstandsfähigkeit zu verbessern. Um sich auf Chaosexperimente vorzubereiten, müssen die Teams Hypothesen über das Verhalten ihrer Systeme entwickeln, wodurch sie besser verstehen, wie Systeme aufgebaut sind und wie sie funktionieren. Bei dieser Vorbereitung werden oft mentale Lücken, architektonische Erkenntnisse und betriebliches Wissen

aufgedeckt, das andernfalls möglicherweise unentdeckt bleiben würde. Chaos Engineering fördert das Verständnis dafür, wie komplexe Systeme auf Ausfälle reagieren, und fördert so mehr Transparenz und Rechenschaftspflicht bei Systemdesign und -management. Je häufiger Chaos Engineering in Ihrem Unternehmen angewendet wird, desto besser ist es darauf vorbereitet, operativ vorzugehen. Chaos Engineering hilft Ihnen dabei, bewährte Methoden für die Entwicklung robuster Anwendungen zu entwickeln, die Komponentenausfälle mit minimalen oder gar keinen Auswirkungen auf die Benutzer überstehen können. Auf diese Weise wird sichergestellt, dass kritische Anwendungen innerhalb der erwarteten Serviceniveaus und der erwarteten Belastbarkeit funktionieren, während gleichzeitig das Wissen des Teams über die eigenen Systeme kontinuierlich erweitert wird.

Wir bereiten uns auf widrige Bedingungen vor

Wenn Sie darauf aufbauen AWS, nutzen Sie verschiedene Arten von Diensten, darunter zonale Dienste wie Amazon Elastic Compute Cloud (Amazon EC2), regionale Dienste wie Amazon Simple Storage Service (Amazon S3), globale Dienste wie AWS Identity and Access Management (IAM), Software-as-a-Service (SaaS) -Dienste von Drittanbietern und lokale Dienste. Jeder Servicetyp macht unterschiedliche Fehlerdomänen verfügbar, die Sie berücksichtigen müssen. Wie bereiten Sie sich auf selbst verschuldete Ereignisse oder auf Ereignisse vor, die von Dritten verursacht werden und auf die Ihr Unternehmen keinen Einfluss hat?

Um besser zu verstehen, wie Ihre Anwendung auf widrige Bedingungen reagieren könnte, können Sie () verwenden AWS Fault Injection Service .AWS FIS AWS FIS ist ein vollständig verwalteter Service für die kontrollierte Durchführung von Fault-Injection-Experimenten. Sie können diesen Service verwenden, um von Ihnen AWS bereitgestellte Szenarien wie Stromausfälle in der Availability Zone und regionsübergreifende Verbindungsprobleme einzufügen, oder Sie können Ihre eigenen Experimente erstellen, indem Sie eine Vielzahl von Fehleraktionen, die vom Service bereitgestellt werden, miteinander verknüpfen. AWS FIS ermöglicht es Ihren Teams, kontinuierlich zu üben und zu lernen, wie ihre Anwendung auf häufig auftretende Fehler reagieren und Fehler beheben würde, sobald sie entdeckt werden.

Üben Sie kontrolliertes Chaos Engineering

Die wichtigsten Prinzipien von Experimenten mit kontrolliertem Chaos sind:

- Beginnen Sie mit einer Umgebung, die Ihrer Produktionsumgebung ähnelt.
- Stellen Sie eine Hypothese auf und beenden Sie die Bedingungen für Ihr Experiment.

- Fangen Sie klein an.
- Üben Sie die Kontrolle über Ihre Chaos-Experimente aus.
- Lege den Umfang der Wirkung fest.
- Informieren Sie sich über Ihre Servicebasis.
- Planen Sie Experimente.
- Korrigieren Sie zuerst und experimentieren Sie dann.
- Überwachen Sie das Experiment genau.
- Lernen Sie aus Ihren Ergebnissen.
- Priorisieren Sie die Ergebnisse, korrigieren Sie sie und überprüfen Sie sie.
- Verbreiten Sie die Erkenntnisse in Ihrem gesamten Unternehmen.

Um Chaos Engineering erfolgreich zu skalieren, müssen Sie Chaosexperimente auf kontrollierte Weise durchführen. Wenn Sie verwenden AWS FIS, können Sie mithilfe von CloudWatchAmazon-Alarmen Stoppbedingungen erstellen. Sie können diese Bedingungen in eine Versuchsvorlage integrieren, um sicherzustellen, dass Experimente gestoppt werden, wenn sie außerhalb der Grenzwerte liegen, und auf ihren letzten bekannten Status zurückgesetzt werden. AWS FIS bietet auch Sicherheitshebel. Wenn Sie diese Hebel betätigen, werden alle laufenden Experimente auf dem Konto in der AWS FIS gestoppt und zurückgesetzt AWS-Region, einschließlich Experimenten mit mehreren Konten, und verhindert, dass neue Experimente gestartet werden. Dadurch wird verhindert, dass Fehler während bestimmter Zeiträume wie Handelszeiten, Verkaufsereignisse oder Produkteinführungen oder als Reaktion auf Integritätsalarme von Anwendungen auftreten. Der Sicherheitshebel bleibt so lange gedrückt, bis er manuell gelöst wird.

Wenn Sie ein Chaosexperiment durchführen, sollten Sie Schutzmaßnahmen festlegen, um unerwünschte Nebenwirkungen auf die Umgebung zu verhindern, insbesondere wenn die Möglichkeit besteht, dass das Experiment Anwendungen beeinträchtigt, die sich in der Produktion befinden. Wenn Sie das Experiment planen, sollten Sie mit etwaigen negativen Auswirkungen rechnen, die es auf andere Anwendungen in der Umgebung haben könnte. Beispielsweise könnten andere Anwendungen fehlerhafte Nachrichten von der Anwendung erhalten, die Teil des Experiments ist, hohe Anforderungszahlen haben oder auf Ressourcenkonflikte stoßen, wenn sie die Infrastruktur gemeinsam nutzen. Dokumentieren Sie diese Risiken und beheben Sie alle bekannten oder inakzeptablen Probleme, bevor Sie das Experiment durchführen.

Erste Schritte mit Chaos Engineering

Bevor Sie ein Experiment durchführen, empfehlen wir Ihnen, einige Grundlagen festzulegen, um das Beste aus Ihren Chaos-Engineering-Praktiken herauszuholen. Zu diesen Grundlagen gehören:

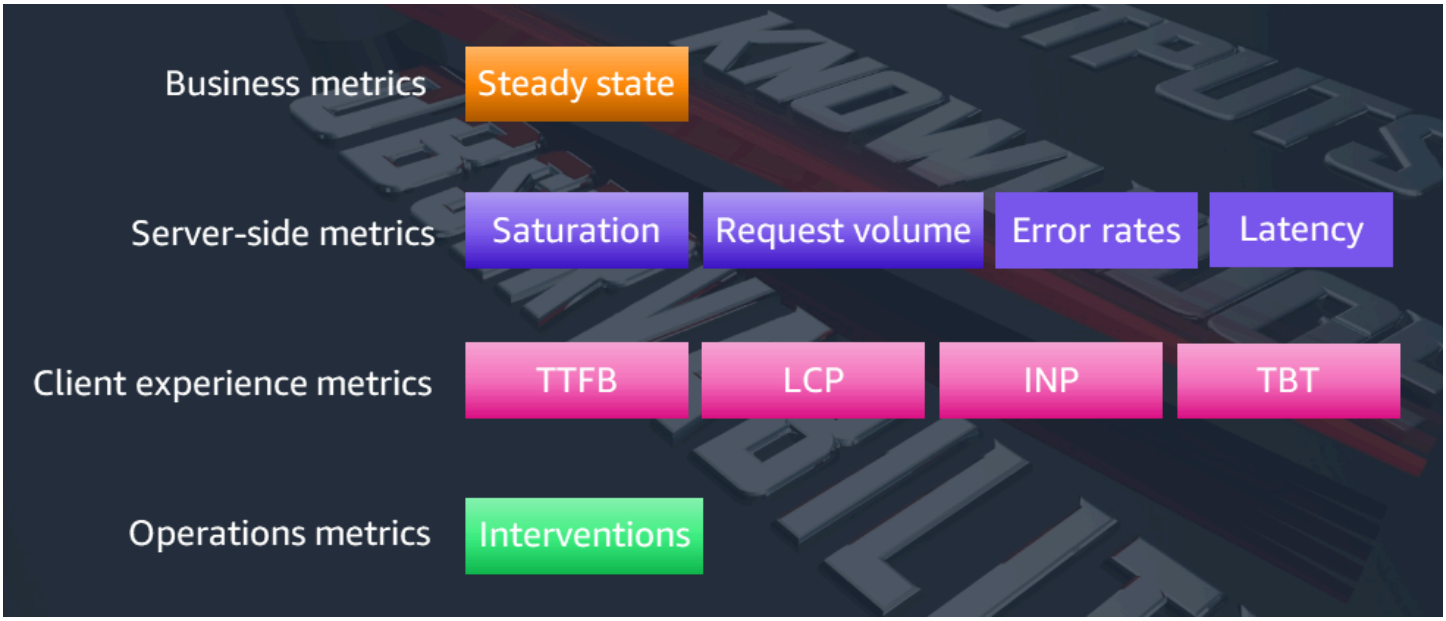
- Beobachtbarkeit (Metriken, Protokollierung, Nachverfolgung von Anfragen)
- Eine Liste von realen Ereignissen oder Fehlern, die Sie untersuchen möchten
- Förderung organisatorischer Resilienz durch Mitwirkung von Führungskräften
- Priorisierung kritischer Ergebnisse auf der Grundlage potenzieller Auswirkungen auf das Geschäft gegenüber neuen Funktionen, die bei der Durchführung von Chaosexperimenten entdeckt werden

Beobachtbarkeit von Chaosexperimenten

Die Beobachtbarkeit, die Metriken, Protokollierung und Nachverfolgung von Anfragen umfasst, spielt beim Chaos Engineering eine Schlüsselrolle. Wenn Sie ein Experiment durchführen, sollten Sie Geschäftskennzahlen, serverseitige Metriken, Kennzahlen zur Kundenerfahrung und Betriebsmetriken verstehen. Ohne Beobachtbarkeit können Sie kein stabiles Verhalten definieren oder ein aussagekräftiges Experiment durchführen, um zu überprüfen, ob Ihre Hypothese über Ihre Anwendung zutrifft.

Kennzahlen

Das folgende Diagramm zeigt die Arten von Metriken, die Sie für Chaosexperimente für verschiedene Arten von Anwendungen verfolgen können.



- Geschäftskennzahlen — Steady State gibt den normalen Betrieb Ihres Systems an und wird durch Ihre Geschäftskennzahlen definiert. Sie kann durch Transaktionen pro Sekunde (TPS), Klick-Streams pro Sekunde, Bestellungen pro Sekunde oder eine ähnliche Kennzahl dargestellt werden. Ihre Anwendung weist einen stabilen Zustand auf, wenn sie erwartungsgemäß funktioniert. Stellen Sie daher sicher, dass Ihre Anwendung fehlerfrei ist, bevor Sie Experimente durchführen. Steady State bedeutet nicht unbedingt, dass ein Fehler keine Auswirkungen auf die Anwendung hat, da ein Prozentsatz der Fehler innerhalb akzeptabler Grenzen liegen kann. Der Steady-State ist Ihr Ausgangswert. Der stabile Zustand eines Zahlungssystems könnte beispielsweise als die Verarbeitung von 300 TPS mit einer Erfolgsquote von 99 Prozent und einer Hin- und Rücklaufzeit von 500 ms definiert werden. Stellen Sie sich den Steady-State visuell wie ein Elektrokardiogramm (EKG) vor. Wenn der stationäre Zustand Ihres Systems plötzlich schwankt, wissen Sie, dass ein Problem mit Ihrem Service vorliegt.
- Server-side Kennzahlen — Um zu verstehen, wie sich Ihre Ressourcen während des Experiments verhalten, benötigen Sie Einblicke in ihre Leistung vor, während und nach dem Experiment. Um die Auswirkungen Ihrer Ressourcen auf zu messen AWS, können Sie Amazon verwenden CloudWatch. CloudWatch ist ein Service, der Anwendungen überwacht, auf Leistungsänderungen reagiert, die Ressourcennutzung optimiert und Einblicke in den Betriebszustand bietet. Während Ihrer Experimente sollten Sie serverseitige Messwerte wie Sättigung, Anforderungsvolumen, Fehlerraten und Latenz erfassen.
- Kennzahlen zum Kundenerlebnis — Bei Aktivierung können Sie echte Benutzermetriken erfassen AWS, indem Sie CloudWatchRUM verwenden, um Benutzeranfragen mithilfe von Tools wie Locust, Grafana k6, Selenium oder Puppeteer zu simulieren. Echte Benutzermetriken sind für

Unternehmen, die Chaos-Engineering-Experimente durchführen, von entscheidender Bedeutung. Durch die Überwachung der Auswirkungen auf reale Benutzer während eines Experiments können sich Teams ein genaues Bild davon machen, wie sich Fehler und Störungen auf Kunden in der Produktion auswirken werden. Zu den wichtigsten Kennzahlen für das Kundenerlebnis gehören Time to First Byte (TTFB), Largest Contentful Paint (LCP), Interaction to Next Paint (INP) und Total Blocking Time (TBT).

- Betriebskennzahlen — Interventionen messen, wie erfolgreich Sie Fehler auf automatisierte Weise beheben — zum Beispiel die erfolgreiche Latenz von Client-Anfragen während eines Neustarts von Pods, Tasks oder EC2-Instances mit Mechanismen wie Replikationscontroller oder automatischer Skalierung. Die Möglichkeit, bei einem Fehler automatisch einzugreifen, steht in direktem Zusammenhang mit einer guten Benutzererfahrung. Es ist von entscheidender Bedeutung zu verstehen, ob sich diese Minderungsmechanismen im Laufe der Zeit verändern. Indem Sie Kennzahlen sowohl für erfolgreiche als auch für fehlgeschlagene automatisierte Abhilfemaßnahmen definieren, erstellen Sie Wegweiser, anhand derer Sie potenzielle Regressionen in Ihrem System identifizieren können.

Protokollierung

Die zentrale Protokollierung ist entscheidend, um den Status der Komponenten Ihrer Anwendung vor, während und nach einem Chaosexperiment zu verstehen. Wir empfehlen Ihnen, Protokolle von all Ihren Anwendungskomponenten zu sammeln, um eine konsolidierte Übersicht darüber zu erhalten, was die einzelnen Komponenten zum Zeitpunkt der Installation des Experiments getan haben. Auf diese Weise erhalten Sie ein klares Bild vom gesamten Ablauf des Experiments.

Anfragenachverfolgung

Mit der Anforderungsverfolgung können Sie den Ablauf jeder einzelnen Anfrage über die Komponenten Ihrer Anwendung hinweg beobachten, um ein umfassendes Verständnis der Auswirkungen zu erhalten, die der injizierte Fehler auf das System und seine Abhängigkeiten hat. Durch die Rückverfolgung der Anfragen können Sie erkennen, wie sich der Fehler auf verschiedene Dienste und Komponenten ausbreitet, sodass Sie den Umfang der Störung besser einschätzen können. Um Ihre Anfragen nachzuverfolgen AWS, können Sie verwenden. AWS X-Ray

Misserfolgszenarien zur Injektion von Chaosexperimenten

Das Ziel, häufig auftretende Fehler in Ihre Anwendung einzufügen, besteht darin, zu verstehen, wie die Anwendung auf diese unerwarteten Ereignisse reagiert. Auf diese Weise können Sie Abhilfemaßnahmen einrichten und Ihr System widerstandsfähig gegen solche Fehler machen. Darüber hinaus sollten Sie Chaos Engineering verwenden, um historische Ausfallszenarien nachzuspielen, um sicherzustellen, dass Ihre Abwehrmechanismen immer noch wie erwartet funktionieren und sich im Laufe der Zeit nicht verändert haben.

Berücksichtigen Sie bei der Planung Ihrer Experimente zur Chaos-Engineering-Technik die folgenden Ereignisse.

Fehlermodus	Description
Beeinträchtigung des Servers	Starten Sie EC2-Instances neu, löschen Sie Kubernetes-Pods oder Amazon Elastic Container Service (Amazon ECS) -Aufgaben , um zu verstehen, wie Ihre Anwendung auf solche Abstürze reagiert.
API-Fehler	Fügen Sie Fehler in AWS und Ihre eigenen Service-APIs ein, um das Anwendungsverhalte n zu verstehen.
Probleme mit dem Netzwerk	Führen Sie Latenz oder Überlastung ein oder blockieren Sie Verbindungen, um reale Netzwerkprobleme nachzuahmen.
AWS Beeinträchtigung der Verfügbarkeitszone	Spielen Sie die Beeinträchtigung einer gesamten Availability Zone erneut ab, um die zonenübergreifende Wiederherstellung zu überprüfen.
AWS-Region Beeinträchtigung der Konnektiv ität	Führen Sie eine Netzwerkbeeinträchtigung erneut durch AWS-Regionen , um zu überprüfe n, wie Ressourcen in der sekundären Region auf ein solches Ereignis reagieren.

Fehlermodus	Description
Datenbankausfälle	Führen Sie ein Failover von Datenbank replikaten durch, beschädigen Sie Daten oder machen Sie Datenbankinstanzen unerreich bar, um mehr über die Auswirkungen auf Ihre Anwendungs- und Wiederherstellungsstrategie n zu erfahren.
Pause bei der Datenbank- und Amazon S3 S3-Replikation	Unterbrechen Sie die Datenbank- oder Amazon Simple Storage Service (Amazon S3) -Replikat ion über Availability Zones hinweg oder AWS-Regionen um die Auswirkungen nachgelagerter Anwendungen zu verstehen.
Verschlechterung des Speicherplatzes	Halten Sie an I/O, entfernen Sie Amazon Elastic Block Store (Amazon EBS) -Volumes oder löschen Sie Dateien, um die Haltbarkeit und Wiederherstellung der Daten zu überprüfe n.
Beeinträchtigung der Abhängigkeit	Reduzieren oder verschlechtern Sie die Leistung der nachgelagerten oder vorgelage rten Dienste, auf die Sie angewiesen sind, einschließlich der Dienste von Drittanbietern, um den gesamten Ablauf und die Auswirkun gen auf Ihre Kunden zu verstehen.
Der Verkehr nimmt zu	Generieren Sie Spitzen im Benutzerdatenverke hr, um die Funktionen der automatischen Skalierung zu testen und herauszufinden, wie sich die Kaltstartzeit auf den allgemeinen Anwendungsstatus auswirken kann.

Fehlermodus	Description
Erschöpfung der Ressourcen	Maximieren Sie den CPU-, Arbeitsspeicher- und Festplattenspeicher, um zu überprüfen, ob Ihre Anwendung ordnungsgemäß beeinträchtigt wird.
Kaskadierende Fehler	Initiieren Sie primäre Ausfälle, die sich wiederum auf nachgelagerte Anwendungen und Komponenten auswirken.
Schlechte Bereitstellungen	Führen Sie problematische Änderungen oder Konfigurationen durch, um die Rollback-Mechanismen zu überprüfen.

Förderung organisatorischer Resilienz

Chaos Engineering bietet den größten Nutzen, wenn es in Ihrem gesamten Unternehmen angewendet wird. Wir empfehlen Ihnen, mit einem Sponsor aus der Geschäftsleitung zusammenzuarbeiten, der Ihnen helfen kann, Resilienzziele in Ihrem gesamten Unternehmen festzulegen, Ängste, Unsicherheiten und Zweifel in Bezug auf den Bereich zu beseitigen und den Transformationsprozess einzuleiten, sodass Resilienz in die Verantwortung aller fällt.

Um das Geschäftsszenario des Aufbaus einer Chaos-Engineering-Praxis zu unterstützen, sollten Sie Ihre kritischen Geschäftsprojekte mit Chaos-Engineering-Maßnahmen verknüpfen. Wenn Sie Resilienz zu einem Vorteil und einer treibenden Kraft für die Beschleunigung machen, können Sie Ihren Erfolg im Laufe der Zeit verfolgen. Beginnen Sie mit der Anzahl kritischer Vorfälle pro Monat oder pro Quartal, der durchschnittlichen Wiederherstellungszeit und den Auswirkungen, die diese Vorfälle auf Ihre Kunden und Ihr Unternehmen hatten. Setzen Sie sich gemeinsam mit Ihren Teams das Ziel, die Anzahl der Vorfälle über einen Zeitraum von 6 bis 12 Monaten zu reduzieren, da als Reaktion auf Chaos-Engineering-Experimente Verbesserungen in Ihren gesamten Anwendungsstapeln vorgenommen werden.

Messen Sie, ob sich Vorfälle stark wiederholen. Nehmen wir zum Beispiel an, ein abgelaufenes TLS-Zertifikat führt zu Ausfallzeiten, da Clients keine vertrauenswürdige Verbindung aufbauen können. Wenn in einem Jahr mehrere Vorfälle auftreten, weil mehrere TLS-Zertifikate ablaufen, können Sie ein Experiment mit dem Ablauf eines TLS-Zertifikats durchführen und sicherstellen, dass Ihre Teams

Benachrichtigungen erhalten oder in der Lage sind, solche Probleme automatisch zu beheben. Auf diese Weise können Sie sicherstellen, dass Sie gegen solche Fehler resistent sind.

Um den Fortschritt beim Chaos Engineering im Laufe der Zeit zu verfolgen, sollten Sie die folgenden Kennzahlen erfassen, um den Wert von Chaos Engineering über den gesamten Lebenszyklus einer Anwendung hinweg zu verdeutlichen:

- Geringere Vorfallrate — Verfolgen Sie die Anzahl der Produktionsvorfälle im Laufe der Zeit und korrelieren Sie diese Zahl mit der Einführung von Chaos Engineering. Es wird davon ausgegangen, dass die Zahl der schwerwiegenden Vorfälle sinken wird.
- Verbesserte mittlere Lösungszeit (MTTR) — Berechne die durchschnittliche Zeit, die zur Behebung von Vorfällen benötigt wird, und verfolge diese Daten, um festzustellen, ob sie sich mit der Zeit durch Chaos Engineering verbessern.
- Höhere Anwendungsverfügbarkeit — Verwenden Sie Metriken auf Service-Ebene, um Verbesserungen der Verfügbarkeit aufzuzeigen, wenn die Ausfallsicherheit von Anwendungen durch Chaosexperimente zunimmt.
- Schnellere Markteinführung — Chaos Engineering gibt Ihnen die nötige Sicherheit, um innovative Angebote schneller auf den Markt zu bringen, da Sie wissen, dass Ihre Anwendungen robust sind. Verfolgen Sie die zunehmende Geschwindigkeit der Produktveröffentlichung.
- Senkung der Betriebskosten — Ermitteln Sie, ob Indikatoren wie Warnmeldungen, Betriebsauslastung und manueller Aufwand bei der Verwaltung von Anwendungen aufgrund von Chaos-Praktiken abnehmen.
- Stärkung des Vertrauens — Befragen Sie Entwickler, Site Reliability Engineers (SREs) und andere technische Mitarbeiter, um festzustellen, ob Chaos Engineering ihr Vertrauen in die Ausfallsicherheit von Anwendungen gestärkt hat. Wahrnehmungen sind wichtig.
- Verbessertes Kundenerlebnis – Connect Chaos Engineering mit positiven Ergebnissen für Kunden, wie z. B. weniger Serviceunterbrechungen, Rollbacks und Ausfällen.

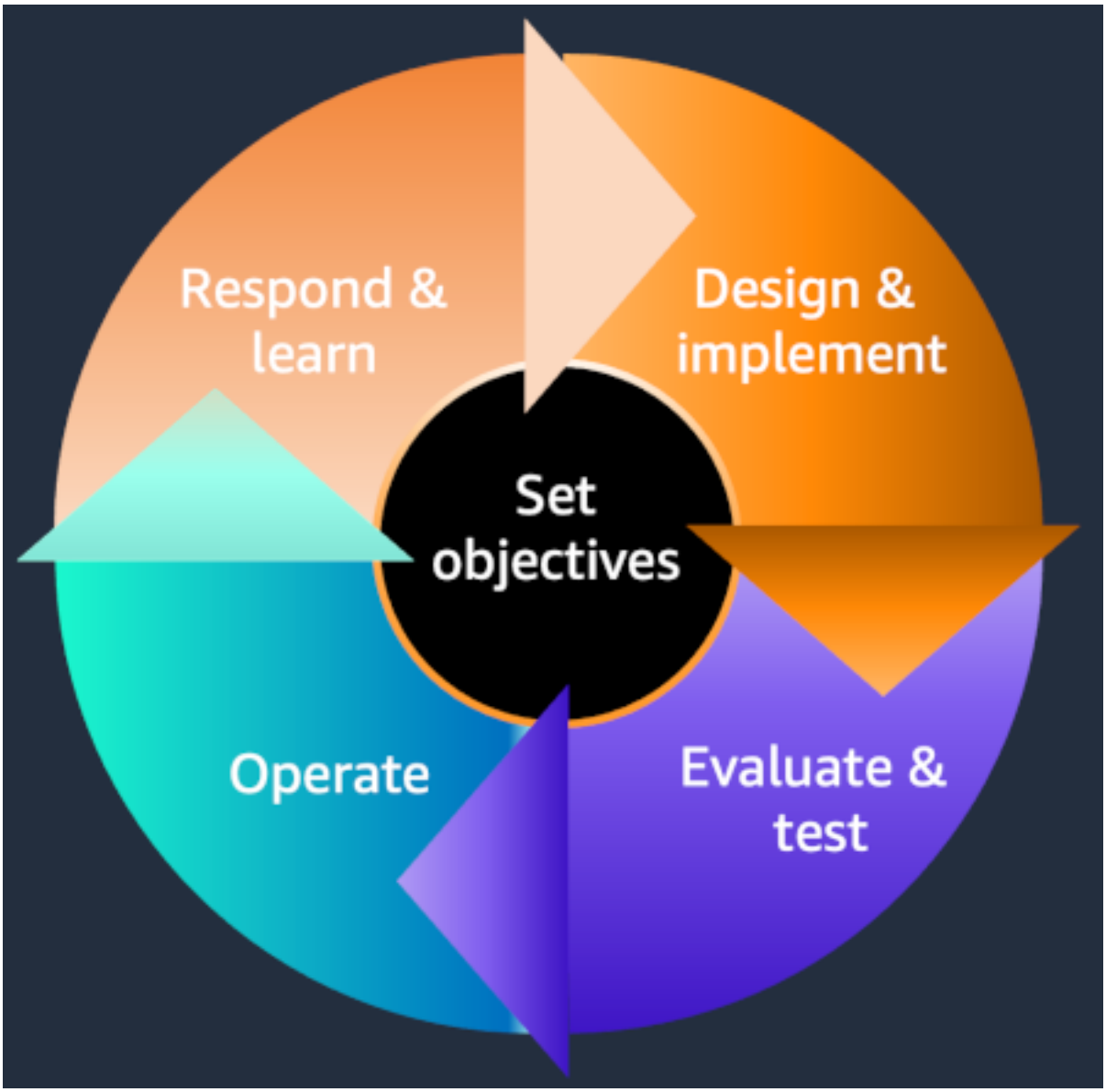
Priorisierung von Problembehebungen

Wenn Sie Chaosexperimente durchführen, werden Sie wahrscheinlich Bereiche identifizieren, in denen Verbesserungspotenzial besteht, in denen die Anwendung nicht wie vorgesehen funktioniert. Die Behebung solcher Probleme wird zu einer Aufgabe in Ihrem Backlog werden, der zusammen mit anderen Arbeiten, wie der Entwicklung von Funktionen, Priorität eingeräumt werden muss. Wir empfehlen Ihnen, sich Zeit für diese Verbesserungen zu nehmen, um future

Ausfälle zu vermeiden. Erwägen Sie, diese Erkenntnisse und Problembehebungsaufgaben nach dem Ausmaß ihrer möglichen Auswirkungen zu priorisieren. Erkenntnisse, die sich direkt auf die Belastbarkeit oder Sicherheit Ihrer Anwendung auswirken, sollten Vorrang vor neuen Funktionen haben, um negative Auswirkungen auf Kunden zu vermeiden. Wenn das Team Schwierigkeiten hat, Abhilfemaßnahmen der Entwicklung von Funktionen vorzuziehen, sollten Sie sich an Ihren Sponsor in der Geschäftsleitung wenden, um sicherzustellen, dass die Prioritäten auf der Grundlage der Risikobereitschaft des Unternehmens festgelegt werden.

Implementierung von Chaos Engineering auf AWS

Chaos Engineering ist Teil der Evaluierungs- und Testphase des AWS Resilienz-Lebenszyklus, wie in der folgenden Abbildung dargestellt. Verteilte Anwendungen können nicht isoliert von anderen Anwendungen oder Clients betrieben werden. Wir empfehlen daher, den gesamten Resilienz-Lebenszyklus zu überprüfen. Verteilte Anwendungen ändern sich ständig, da sich das Netzwerk weiterentwickelt, Upstream- und Downstream-Anwendungen Veränderungen unterliegen und die Client-Nutzung sich im Laufe der Zeit ändert.



Um zu verstehen, wie sich diese Änderungen an Ihrer Anwendung auf deren Ausfallsicherheit auswirken können, sollten Sie Chaos Engineering zu einem Teil Ihres täglichen Betriebs machen. Sie können Chaos-Experimente auf unterschiedliche Weise implementieren:

- Ad hoc — Sie können Chaosexperimente als einmalige Experimente durchführen, um ein bestimmtes Problem oder eine bestimmte Frage zu lösen.
- Chaos-Spieltage — Dabei handelt es sich um strukturierte und wiederkehrende Ereignisse, mit denen die Zuverlässigkeit und Belastbarkeit einer Anwendung überprüft werden soll. Der Zweck eines Chaos-Spieltages besteht darin, potenzielle Resilienzprobleme oder -mängel bei Personen, Prozessen und Technologien zu identifizieren und die Prozesse und Verfahren zur Identifizierung, Minderung und Reaktion auf Vorfälle zu üben.
- Chaos-Pipeline — Bei der kontinuierlichen Integration und kontinuierlichen Bereitstellung (CI/CD) geht es darum, neue Funktionen zu entwickeln und diese sicher in allen Umgebungen einzusetzen. Um Chaos-Engineering-Experimente durchzuführen, erstellen Sie eine Chaos-Pipeline, die von Ihrer CI/CD Pipeline getrennt ist. Um zu verstehen, warum das so ist, nehmen wir an, dass Sie Ihrer CI/CD Pipeline ein einziges Chaos-Engineering-Experiment hinzufügen möchten, das zu einem zunehmenden Paketverlust für nachgeschaltete Komponenten führt. Dieses Experiment wird dreimal ausgeführt und es dauert jeweils 5 Minuten, bis es abgeschlossen ist. Der Paketverlust steigt mit jedem Durchlauf von 10 auf 20 bis 30 Prozent, und das Experiment dauert insgesamt 15 Minuten. Wenn Sie 100 parallel Bereitstellungen haben, müssen Sie 1500 Minuten warten, bis ein einzelnes Experiment abgeschlossen ist. Wenn Sie 10 Experimente durchführen müssen, wären die Auswirkungen für Ihre Entwickler unerträglich. Im großen Maßstab benötigt Chaos Engineering eine eigene Pipeline, die es Ihnen ermöglicht, Experimente parallel zu Ihrem Software Development Lifecycle (SDLC) -Prozess durchzuführen.
- Bereitstellungen auf den Kanarischen Inseln — Die Kanarischen Inseln bieten eine Testumgebung für Chaos-Experimente. Indem Sie einen kleinen Teil des Datenverkehrs an einen Canary-Dienst weiterleiten oder Methoden wie Verkehrsspiegelung oder -wiedergabe verwenden, können Sie neue Infrastruktur- oder Codeänderungen verifizieren, ohne dass dies Auswirkungen auf Ihr stabiles Produktionssystem hat. Sie können Experimente gegen den Kanarienvogel durchführen und bei Bedarf Fehler einschleusen, da Sie den Umfang der Auswirkungen auf den Endbenutzer einschränken können.
- Geplante Experimente — Sie können Experimente planen, um vorhersehbare Wiederherstellungsmechanismen für Ihre Anwendung zu überprüfen. Verwenden Sie geplante Experimente, um allgemein bekannte Ereignisse wiederzugeben und zu erfassen, wie sich Ihre Systeme nach Ereignissen wie dem Beenden einer EC2-Instance hinter einer automatischen

Skalierungsgruppe, dem Entfernen eines Kubernetes-Pods oder dem Löschen einer Amazon ECS-Aufgabe erholen können.

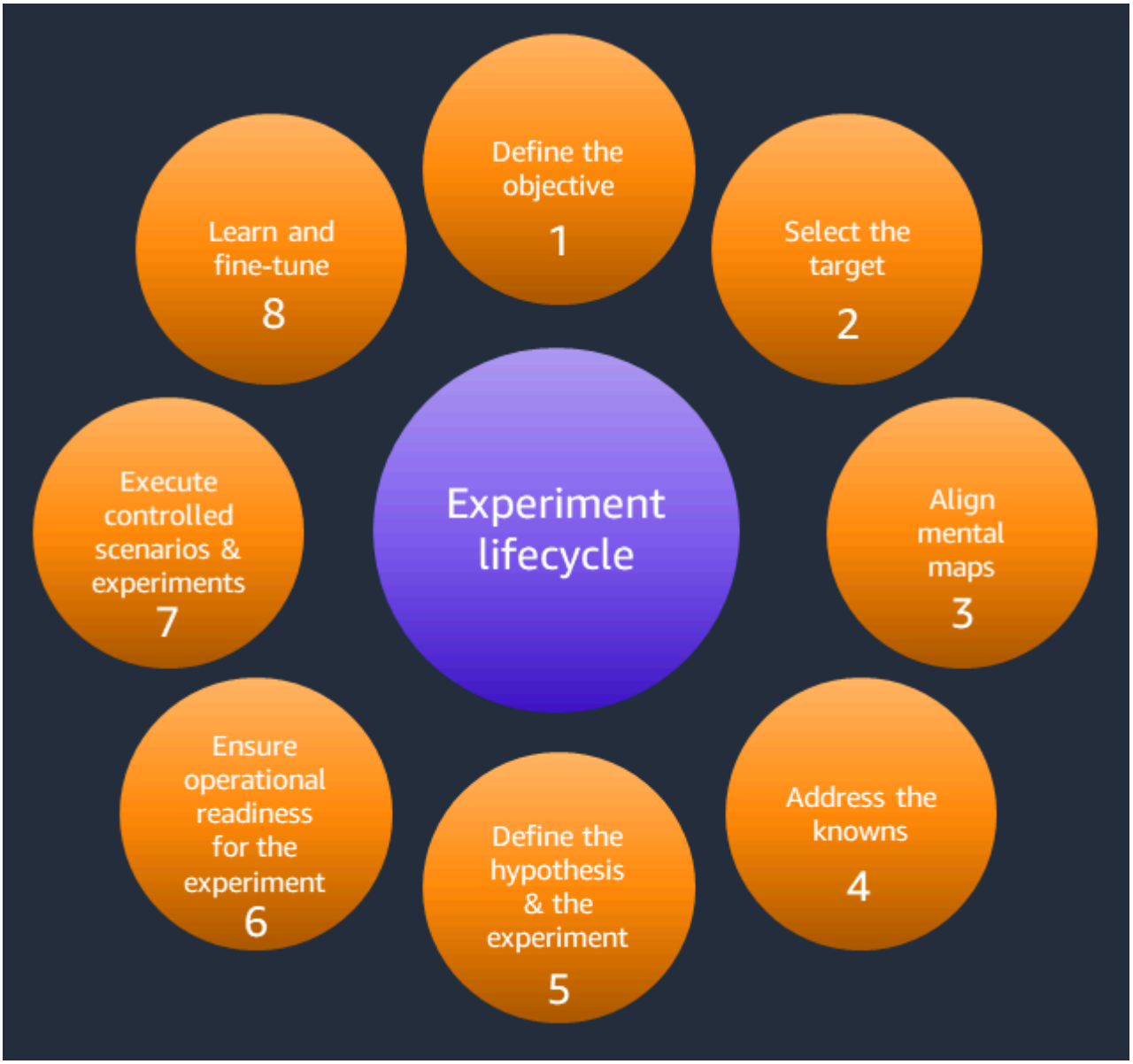
Kontinuierlicher Lebenszyklus von Experimenten in Chaos-Technik

Wie im vorherigen Abschnitt beschrieben, können Sie Chaos-Engineering-Experimente auf unterschiedliche Weise durchführen. In allen Fällen liegt der Schlüssel zum Aufbau eines aussagekräftigen Chaosexperiments darin, die Anwendung, historische Vorfälle und umgesetzte Abhilfemaßnahmen zu verstehen und genau zu verstehen, welche Bereiche untersucht werden müssen, z. B. Widerstandsfähigkeit oder Sicherheit. Ihr Wissen über die Anwendung hilft Ihnen, eine Hypothese über die potenziellen Schwächen der Anwendung zu formulieren und zu verstehen, wie sie den Fehler erkennt, behebt und repariert, wenn der Fehler auftritt.

Der Lebenszyklus eines Chaos-Experiments umfasst die folgenden Schritte:

1. Definieren Sie das Ziel des Experiments.
2. Wählen Sie die Zielanwendung aus.
3. Richten Sie mentale Landkarten aus.
4. Beheben Sie die bekannten Probleme mit Ihrer Anwendung.
5. Definieren Sie die Hypothese und das Experiment.
6. Stellen Sie die Betriebsbereitschaft für das Experiment sicher.
7. Führen Sie kontrollierte Szenarien und Experimente durch.
8. Lernen Sie aus dem Experiment und optimieren Sie es.

Diese Schritte werden in der Abbildung veranschaulicht und in den folgenden Abschnitten erörtert.



Definieren Sie Ziele und legen Sie Erwartungen fest

Stellen Sie vor jedem Experiment sicher, dass Ihre Ziele und Erwartungen spezifisch, messbar, erreichbar, relevant und zeitgebunden sind. Definieren Sie Folgendes klar:

- Identifizieren Sie potenzielle Ausfälle oder Schwächen in Systemen und Diensten, um zu verstehen, wie sie sich auf Benutzer auswirken könnten. Dazu gehört die Identifizierung möglicher Fehlermodi wie Serverabstürze, Netzwerkausfälle oder Softwarefehler und die Bewertung ihrer potenziellen Auswirkungen auf die Gesamtleistung und Zuverlässigkeit des Systems.

- Quantifizieren Sie die Auswirkungen von Ausfällen, indem Sie wichtige Risikoindikatoren (KRIs) für Ihre Systeme und Dienste definieren. Dazu gehört auch die Messung der Auswirkungen von Ausfällen, wenn Kennzahlen wie Latenz, Durchsatz und Fehlerraten von ihren stabilen Werten abweichen. Wenn Sie die Auswirkungen solcher Abweichungen verstehen, können Sie Maßnahmen zur Minderung von Ausfällen auf der Grundlage von Geschäftsrisiken priorisieren.
- Entwickeln und überprüfen Sie Strategien zur Minderung oder Vermeidung von Ausfällen. Dazu gehört die Identifizierung potenzieller Lösungen, wie Redundanz, Fehlerkorrektur oder Fallback-Strategien, und das Testen ihrer Wirksamkeit in einer kontrollierten Umgebung. Durch die Überprüfung dieser Strategien können Sie sicherstellen, dass Sie Ausfälle wirksam verhindern oder abmildern und sie vertrauensvoll in Ihren Produktionssystemen einsetzen können.
- Verbessern Sie die Prozesse für die Reaktion auf Vorfälle und die Notfallwiederherstellung. Durch die Wiederholung von Ausfällen in einer kontrollierten Umgebung können Sie Prozesse zur Reaktion auf Vorfälle testen, potenzielle Engpässe oder Lücken identifizieren und die Notfallwiederherstellungsverfahren verfeinern. Auf diese Weise können Sie sicherstellen, dass Sie auf unerwartete Ausfälle schnell und effektiv reagieren können.

Wählen Sie die Zielanwendung

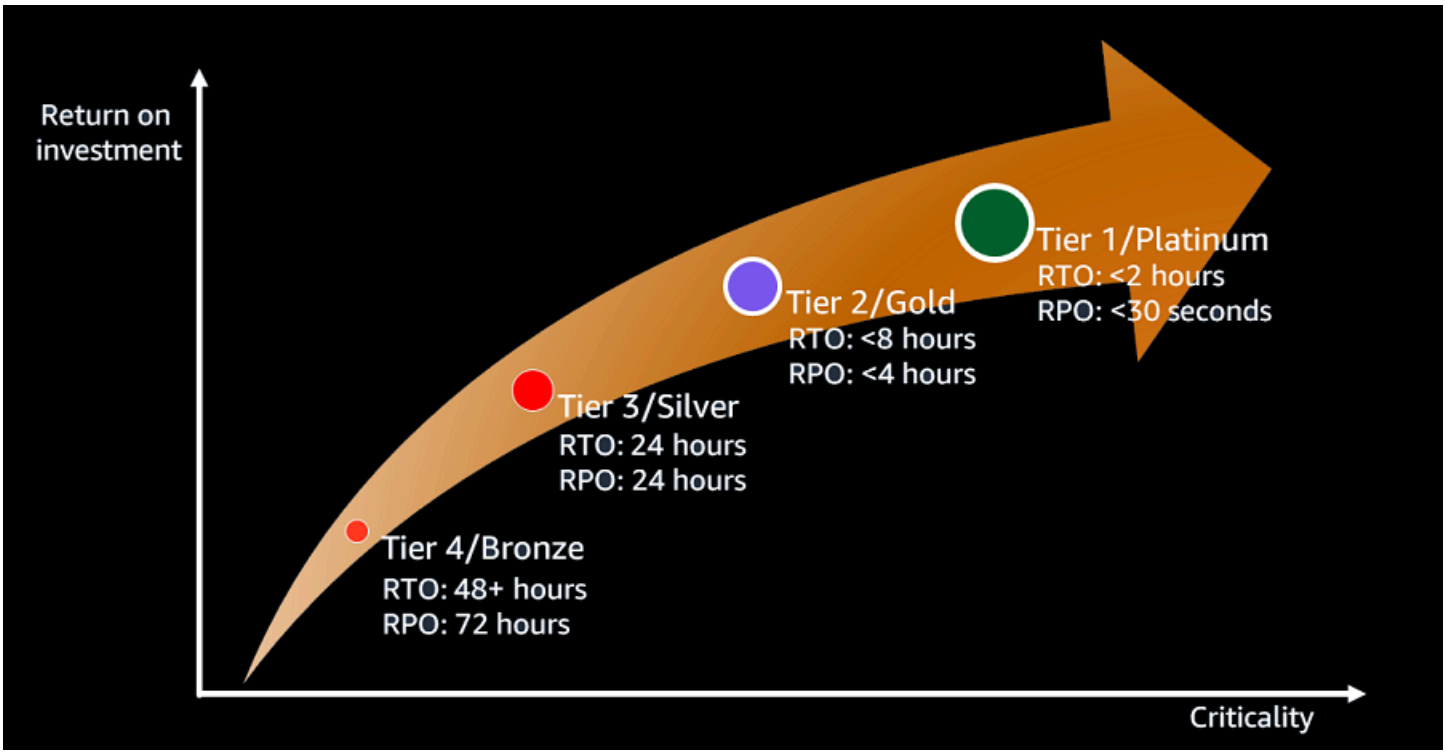
Chaos Engineering ist eine leistungsstarke Technik, erfordert jedoch eine sorgfältige Priorisierung, um den Nutzen zu maximieren. Bei der Entscheidung, worauf sich die Bemühungen im Bereich Chaos Engineering konzentrieren sollen, sollten Sie zunächst die kritischen Services Ihres Unternehmens berücksichtigen. Bitten Sie Ihre Teams, die Phasen des Softwareentwicklungszyklus zu durchlaufen und zunächst Fehler in die Testumgebungen einzuschleusen. Business-critical Anwendungen stehen in direktem Zusammenhang mit Umsatz, Kundenerlebnis und Kernbetrieb. Chaosexperimente mit diesen Diensten können Sicherheitslücken aufdecken, die das Unternehmen — und möglicherweise ganze Märkte — ernsthaft beeinträchtigen können, wenn sie nicht behoben werden. Konzentrieren Sie sich beispielsweise zunächst auf kundenorientierte Dienstleistungen wie Handelssysteme oder Ordersysteme. Die Priorisierung dieser zentralen Dienste bietet den größtmöglichen Schutz pro Zeitinvestition.

Schauen Sie sich nach wichtigen Diensten grundlegende Komponenten wie Datenbanken, Nachrichtenwarteschlangen, Netzwerke und Shared Services-APIs an. Diese können in Ihrer gesamten Organisation als gemeinsam genutzte Komponenten oder Dienste verwendet werden, sodass ihr Ausfall zu weitreichenden Problemen führen kann. Wenn die Widerstandsfähigkeit von Infrastrukturdiensten bestätigt wird, kann man sich darauf verlassen, dass sie die ihnen übergeordneten abhängigen Anwendungen nicht lahmlegen. Beispielsweise verrät ein Chaos

Engineering-Experiment, bei dem ein Kafka-Cluster außer Betrieb genommen wird, viel über die Fehlertoleranz nachgeschalteter Anwendungen. Obwohl die Systeminfrastruktur nicht direkt auf den Kunden ausgerichtet ist, ist sie ein vorrangiges Ziel der Chaos-Technik.

Vergessen Sie nicht, die mentalen Lücken zwischen Mitarbeitern, Prozessen, Anlageninformationen und Abhängigkeiten von Drittanbietern abzubilden, da diese zu erheblichen Störungen führen können, wenn sie nicht mit den Zielen Ihrer Organisation im Hinblick auf die Belastbarkeit im Einklang stehen. Weitere Informationen zur Messung des ROI von Chaos Engineering finden Sie unter Quantifizierung des ROI von Chaos Engineering im Strategiedokument Investieren in Chaos Engineering als strategische Notwendigkeit.

Das folgende Diagramm zeigt die Investitionsrendite, die sich aus der Durchführung von Chaosexperimenten für verschiedene Serviceebenen ergibt.



Mental Maps aufeinander abstimmen (Anwendungserkennung)

Wenn Sie Ad-hoc-Experimente oder Spieltage durchführen, beginnen Sie den Prozess der Anwendungserkennung mit einer Whiteboard-Sitzung, die sich darauf konzentriert, die Details Ihrer Anwendung zu skizzieren. (Wenn Sie die Experimente in der Chaos-Pipeline durchführen, haben Sie diese mentale Landkarte bereits angepasst, indem Sie die Zielanwendung definiert haben.) Ein guter Ansatz, um mentale Lücken zu verstehen, besteht darin, das jüngste Teammitglied zuerst ein

Diagramm der Anwendung zeichnen zu lassen und dann die erfahreneren Mitarbeiter zu bitten, das Diagramm schrittweise zu erweitern. Auf diese Weise werden alle Verständnislücken zwischen den Erfahrungsstufen aufgedeckt.

Das Diagramm sollte sowohl direkte Upstream- als auch Downstream-Abhängigkeiten der Anwendung sowie alle wichtigen Integrationen von Drittanbietern darstellen. Stellen Sie sicher, dass der erwartete Ablauf einer Anfrage durch die Anwendung abgestimmt ist. Stellen Sie die wichtigsten Arbeitsabläufe und Benutzerabläufe fest, um Klarheit darüber zu gewinnen, wie Kunden die Anwendung verwenden. Erwägen Sie die Verwendung eines Sequenzdiagramms, um diese Informationen zu erfassen.

Nach dieser gemeinsamen Sitzung sollte das Team über ein gemeinsames mentales Modell der Anwendung, ihrer kritischen Abhängigkeiten und ihrer Überwachungsmöglichkeiten sowie über ein Verständnis der Risiken verfügen, um eine fundierte Entscheidung treffen zu können, ob ein potenzielles Chaosexperiment fortgesetzt oder abgesagt werden soll.

Beheben Sie die bekannten Probleme mit Ihrer Anwendung

Experimente im Bereich Chaos Engineering sind darauf ausgelegt, proaktiv Defekte in einer Anwendung aufzudecken. Indem Sie Fehler wie erhöhte Latenz, Serverneustarts oder Leistungseinbußen in der Availability Zone einbeziehen, können Sie überprüfen, ob Ihre Anwendung realistische Störungen toleriert. Dieser Prozess setzt jedoch ein gewisses Maß an Stabilität und Integrität der Zielanwendung voraus. Wenn Chaosexperimente mit einer bereits problematischen Anwendung ausgeführt werden, besteht die Gefahr, dass tiefere Probleme verschleiert werden.

Bevor Teams Chaos Engineering durchführen, sollten sie alle bekannten Fehler, Bugs und Leistungsprobleme in ihrer Anwendung beheben.

Definieren Sie die Hypothese und das Experiment

Vorfälle in der Vergangenheit, die zu Störungen Ihrer Anwendung oder anderer Anwendungen in Ihrem Unternehmen geführt haben, können als hervorragende Quellen für Ideen für Chaos-Experimente dienen. Wurden frühere Ausfälle beispielsweise durch Konfigurationsfehler oder fehlende Resilienzmuster ausgelöst? Die Überprüfung der Vorfälle und die Wiederholung der Grundursachen dieser realen Misserfolge durch Chaosexperimente sind ein wirksames Mittel, um Widerstandsfähigkeit gegen ähnliche Probleme in der future zu entwickeln.

Eine weitere wertvolle Quelle für Experimentkonzepte können direkt von den Ingenieuren, Architekten und Betreibern stammen, die mit einer Anwendung am besten vertraut sind. Wenn Sie Teammitgliedern die Möglichkeit geben, hypothetische Ausfallszenarien einzureichen, von denen sie glauben, dass sie die Anwendung erheblich stören könnten, können Sie Ideen sammeln, die auf Insiderwissen basieren. Das Anwendungsteam kann dann bewerten, welches dieser vorgeschlagenen Szenarien die größten potenziellen Auswirkungen haben oder die größten unbekannten Risiken bergen könnte. Die gezielte Ausrichtung von Chaosexperimenten für solche risikoreichen, weniger verstandenen Szenarien kann wichtige Erkenntnisse liefern und future Probleme verhindern.

Eine dritte Quelle für Ideen ist die Durchführung von Resilienzmodellen, um die Bedingungen zu antizipieren, die zu identifizierten Geschäftsverlusten führen würden. Einige Übungen zur Resilienzmodellierung basieren auf einem komponentenbasierten Ansatz zur Erstellung eines Resilienzmodells, während andere einen systembasierten Ansatz verfolgen. Bei einem komponentenbasierten Ansatz wird die Frage gestellt: „Was passiert, wenn Komponente X extrem belastet ist oder ausgefallen ist?“ Das Team, das das Resilienzmodell entwickelt, spekuliert dann über die Auswirkungen eines solchen Szenarios auf die breitere Anwendung und identifiziert die derzeit geltenden Überwachungs- und Präventivmaßnahmen, um die Auswirkungen des Szenarios zu erkennen und abzuschwächen. Alternativ folgt ein systembasierter Ansatz einem Top-down-Prozess, um einen unerwünschten Zustand der Anwendung hervorzuheben — z. B. „Die Web-Storefront zeigt veraltete Lagerbestände“ — und fordert das Anwendungsteam auf, vorherzusehen, welcher Zustand oder welche Bedingungen dazu führen würden, dass sich die Anwendung auf diese Weise verhält.

Stellen Sie die Betriebsbereitschaft für das Experiment sicher

Sie benötigen quantifizierbare Indikatoren, um die Auswirkungen ungünstiger Bedingungen auf die Anwendung und ihr Verhalten zu messen, wie bereits im Abschnitt zur Beobachtbarkeit beschrieben. Wenn Sie das Verhalten der Anwendung messen können, können Sie feststellen, ob und in welchem Ausmaß sich die ungünstigen Bedingungen auf die Anwendung ausgewirkt haben.

Der beste Weg, um zu verstehen, ob es Auswirkungen auf Ihre Anwendung gibt, besteht darin, ihren stationären Zustand zu messen. Steady State misst, wie der normale Betrieb aussieht, und entspricht in der Regel den Geschäfts- und Kundenerfahrungsindikatoren für eine bestimmte Anwendung. Bevor Sie mit dem nächsten Schritt fortfahren, stellen Sie sicher, dass Sie über die erforderlichen Beobachtungsmöglichkeiten verfügen, um die Auswirkungen nachvollziehen zu können, und über Rollback-Mechanismen für den Fall, dass das Experiment nicht wie erwartet verläuft, bereit sind.

Führen Sie kontrollierte Experimente und Szenarien durch

Wir raten AWS davon ab, ein erstes Chaosexperiment mit einer Anwendung durchzuführen, die sich in der Produktion befindet. Der Zweck eines Chaos-Experiments besteht darin, etwas Neues darüber zu erfahren, wie sich die Anwendung unter stressigen Bedingungen verhält. Das Verhalten der Anwendung kann während des Experiments unvorhersehbar sein, sodass die Durchführung eines Experiments zum ersten Mal in der Produktion Auswirkungen auf die Kunden haben kann. Daher sollten Sie ein anfängliches Chaosexperiment immer in einer Umgebung auf niedrigerer Ebene durchführen, die nur minimale Auswirkungen auf reale Benutzer hat, und dann Ihre Umgebungen wiederholen, nachdem Sie sich vergewissert haben und sicher sind, dass Ihre Anwendung die eingeführten Aktionen aufnehmen, sich an sie anpassen und sich von ihnen erholen kann.

Planen Sie jedes Experiment gründlich, indem Sie ein Dokument verwenden, das die wichtigsten Details enthält, ähnlich dem Dokument zur Versuchsplanung im Anhang. Einige der wichtigsten Bereiche, die berücksichtigt werden sollten, sind die Definition des stationären Zustands, die Hypothese und die Methode der Fehlerinjektion. Planung, Durchführung und Analyse eines Chaosexperiments können in einem einzigen Artefakt zusammengefasst werden.

Nachdem Sie den schriftlichen Plan für das Experiment fertiggestellt haben, bereiten Sie den erforderlichen Code vor, um die geplanten Störungen, die im Dokument beschrieben werden, einzufügen.

Um mögliche Auswirkungen während des Experiments zu erfassen, stellen Sie sicher, dass Beobachtungsmechanismen vorhanden sind. Wenn Sie noch nicht über eine automatisierte Methode zur Erfassung von Versuchsergebnissen verfügen, wie z. B. AWS FIS Experimentberichte, identifizieren Sie die Teammitglieder, die sich während der Ausführung Notizen machen, Screenshots von Dashboards erstellen und das Team durch das Experiment führen werden.

Lernen und verfeinern

Trefft euch nach jedem Experiment als Team, um das Chaos-Experiment noch einmal Revue passieren zu lassen und darüber nachzudenken. Bemühen Sie sich bewusst, eine tadellose Denkweise aufrechtzuerhalten. Ihr Ziel sollte es sein, einen offenen, konstruktiven Dialog zu führen, der sich ausschließlich darauf konzentriert, maximalen Lernerfolg zu erzielen, und nicht Schuldzuweisungen.

Überprüfen Sie zunächst die Steady-State-Definition und die Hypothese für das Experiment. Hat sich die Anwendung wie erwartet verhalten? Gab es irgendwelche Überraschungen, die Annahmen

widerlegten? Erläutern Sie die positiven und schlechten Beobachtungen der Anwendung während des Experiments. Die gesammelten Daten — Metriken, Protokolle, Screenshots usw. — sollten genau die Geschichte erzählen, was passiert ist.

Gehen Sie diese Datenüberprüfung mit Neugier statt mit Urteilsvermögen an und identifizieren Sie Bereiche, in denen das Anwendungsdesign, die Dokumentation, die Überwachung oder andere Funktionen auf der Grundlage der gewonnenen Erkenntnisse verbessert werden können. Diese Aktionspunkte werden als Folgeprojekte erfasst, um die Anwendung widerstandsfähiger zu machen.

Durch diesen untadeligen Ansatz können Sie offen darüber sprechen, was schief gelaufen ist und wie Sie es beheben können. Gehen Sie davon aus, dass alle Beteiligten eine positive Absicht haben, und vertrauen Sie darauf, dass sie auf gute Ergebnisse hingearbeitet haben. Ihr gemeinsames Ziel ist organisatorisches Wachstum und Weiterentwicklung durch kontinuierliches Lernen und Anpassung. Überprüfungen von Chaos-Experimenten, die auf konstruktive und untadelige Weise durchgeführt werden, bieten Ihrem Team einen sicheren Ort, an dem es wertvolle Erkenntnisse gewinnen kann, die Ihre Anwendungen und Ihr Unternehmen auf lange Sicht zuverlässiger und widerstandsfähiger machen. Der Fokus liegt weiterhin auf den Erkenntnissen, nicht auf den Menschen. Um die Erkenntnisse in Ihren Teams zu verbreiten, veröffentlichen Sie den Bericht mit den Testergebnissen an einer zentralen Stelle und bewerben Sie die Ergebnisse, damit andere daraus lernen können.

Skalierung von Chaos Engineering in Ihrem gesamten Unternehmen

Wenn Ihr Unternehmen Chaos Engineering einführt, werden dessen Standardisierung und Implementierung mit Herausforderungen verbunden sein. In den frühen Reifephasen werden verschiedene Teams wahrscheinlich unterschiedliche Tools und Varianten des Chaos Engineering-Prozesses verwenden, der in den vorherigen Abschnitten beschrieben wurde. Gleichzeitig räumen einige Teams trotz der potenziellen Vorteile möglicherweise gar keine Prioritäten ein oder setzen sie gar nicht ein. Die folgenden Abschnitte enthalten Anleitungen zur Bewältigung dieser Herausforderungen.

Insgesamt sollte Ihr Ansatz für Chaos Engineering so konzipiert sein, dass er ein Gleichgewicht zwischen zentraler Führung und dezentraler Beteiligung herstellt. Dieses Gleichgewicht trägt dazu bei, dass Chaos Engineering in den Entwicklungsprozess integriert wird und dass die Erkenntnisse innerhalb Ihres Unternehmens geteilt werden.

Etablierung einer Praxis für Chaos-Engineering

Die Standardisierung der Praxis der Chaos-Technik kann deren Einführung beschleunigen. Der Austausch der Erkenntnisse aus Experimenten zwischen den Teams kann die Rendite von Investitionen in Chaos Engineering erhöhen.

Bauen Sie im Rahmen Ihrer Chaos-Engineering-Praxis ein zentrales Exzellenzzentrum auf oder stellen Sie eine Gruppe von Fachexperten zusammen. Als kleine, zentrale Funktion kann dieses Team in allen Softwareentwicklungs-, Infrastruktur-, Sicherheits- und Geschäftsteams arbeiten und die Standards einhalten, die von diesen Teams verwendet werden. Der Einfachheit halber wird das Center of Excellence das Centralized Practice Team genannt, und Gruppen, die Chaos Engineering anwenden, werden im weiteren Verlauf dieses Leitfadens als Übungsteams bezeichnet.

Rolle des zentralisierten Praxisteams

Das zentralisierte Praxisteam ist für die Entwicklung und Implementierung von Chaos-Engineering-Praktiken im gesamten Unternehmen verantwortlich. Sie arbeiten eng mit den Praxisteams zusammen, um sie bei der Planung und Durchführung von Experimenten zu unterstützen und sicherzustellen, dass die Experimente für das Unternehmen wertvoll sind. Das zentralisierte

Praxisteam berät und unterstützt auch die Entwicklungs-, Infrastruktur- und Sicherheitsteams, um sie bei der Integration von Chaos Engineering in ihre Entwicklungsprozesse zu unterstützen.

Zu den wichtigsten Aufgaben eines zentralisierten Praxisteams für Chaos-Engineering gehören:

- Unterstützung — Eine zentrale Abteilung für Chaostechnik fungiert als Vermittler, um die Praxis der Chaostechnik im Rahmen von Spieltagen und Workshops einzuführen. Sie unterstützen Teams bei der Entwicklung von Chaos Engineering, einschließlich der Auswahl von Fehlerszenarien, der Definition von Hypothesen und der Erstellung von Berichten, die der gesamten Organisation zur Verfügung gestellt werden. Das zentralisierte Praxisteam sollte über die eigenen Schulungsmaterialien verfügen und daran arbeiten, die Übungsteams im Umgang mit Chaos Engineering weiterzubilden.
- Beratung — Das zentralisierte Praxisteam kann auch beratend tätig werden und die Experimente beaufsichtigen, die von den Übungsteams durchgeführt werden. Ihre Erfahrung und ihr Wissen können sicherstellen, dass Experimente einen Mehrwert für das Unternehmen bieten und auf sichere Weise durchgeführt werden. In ähnlicher Weise kann das Team die Durchführung und Nachbesprechung eines Experiments beaufsichtigen, um Personen, die mit Chaos Engineering noch nicht vertraut sind, eine Anleitung zu geben.
- Marketing und Wertverfolgung — Die Vermittlung des geschäftlichen Nutzens von Chaos Engineering ist der Schlüssel zum Erfolg eines solchen Programms. Jedes Team, das an Chaos-Engineering-Experimenten teilnimmt, sollte Daten aus den Experimenten im gesamten Unternehmen sammeln und den Wert der Investition des Unternehmens in Chaos Engineering nachweisen. Dazu gehören die Quantifizierung und Würdigung der Anzahl der Vorfälle, die bei jedem Experiment vermieden wurden, der Ausfallzeiten, die bei einem Fehlschlagen des Experiments entstanden wären, und der Gesamtauswirkungen, die sich auf das Unternehmen ausgewirkt hätten, wenn die Ausfallszenarien in der Produktion aufgetreten wären. Durch die Erfassung und Zentralisierung solcher Daten aus allen Teams und die Bereitstellung der Daten im gesamten Unternehmen kann das zentralisierte Praxisteam den Nutzen verfolgen und beeinflussen, der sich aus der Einführung von Chaos Engineering im gesamten Unternehmen ergibt.
- Standards — Das zentralisierte Praxisteam sollte den Prozess zur Durchführung von Chaos-Experimenten, die Vorlagen für die Planung und Berichterstattung über Experimente und die zur Durchführung von Experimenten verwendeten Tools selbst in die Hand nehmen und verwalten.

  Das zentrale Team sollte die Vorlagen für die Experimentplanung, die Vorlagen für die Versuchsberichte, die Prozessdokumentation und die Materialien für die Durchführung verwalten und verwalten. Die Dokumentation bewährter Verfahren und das Begleitmaterial bieten erfahrenen

Teams Anleitungen zu Themen wie den Leitplanken, mit denen sie die Auswirkungen eines Experiments begrenzen können, wann ein Experiment in der Produktion durchgeführt werden sollte und wie sie Chaos Engineering im Laufe der Zeit weiterentwickeln können. Beispiele für Vorlagen und Ergebnisse finden Sie im Anhang.

Das zentralisierte Praxisteam sollte auch für den Ablauf der Durchführung eines Experiments verantwortlich sein, einschließlich Kommunikation und Eskalation, und darüber, wann und wie vor oder während eines Experiments mit anderen Teams in der Organisation kommuniziert werden soll. In dem Prozess sollte auch dargelegt werden, wann Leitplanken erforderlich sind.

Das zentralisierte Praxisteam sollte auch die wichtigsten Tools für die Durchführung von Chaosexperimenten auswählen und für sich entscheiden (z. B. Tools wie AWS FIS). Die Entscheidung über die Auswahl und Implementierung zusätzlicher Instrumente, wie z. B. Tools zur Lastgenerierung, sollte den Praxisteams überlassen werden. Übende Teams sollten in der Lage sein, den Gesamtprozess und die Tools so anzupassen, dass sie ihren Bedürfnissen am besten entsprechen.

Rolle der praktizierenden Teams

Das zentralisierte Team ist dafür verantwortlich, die gesamte Chaos-Engineering-Strategie voranzutreiben, wohingegen die Übungsteams am Prozess beteiligt sind und für die Entwicklung und Durchführung der Experimente verantwortlich sind. Dadurch wird sichergestellt, dass die Experimente für jedes spezifische Produkt oder jede Dienstleistung relevant sind und dass die Erkenntnisse umsetzbar sind und zur Verbesserung der Zuverlässigkeit und Widerstandsfähigkeit des Produkts angewendet werden können. Das zentralisierte Praxisteam fungiert als Mentor und verantwortlich für die Chaos-Engineering-Standards und -Prozesse der Organisation. Um jedoch zu verhindern, dass das zentralisierte Team zu einem Engpass wird, müssen die einzelnen Übungsteams von der zentralen Praxis lernen, um Chaosexperimente selbst durchzuführen.

Aufbau einer Praxisgemeinschaft

Wir empfehlen Ihnen, nicht nur ein zentralisiertes Team zusammenzustellen, sondern auch eine informelle Gemeinschaft von Praktikern aufzubauen, die sich für Chaos Engineering interessieren. Diese Community bietet eine Plattform für den Austausch von Wissen, bewährten Verfahren und Erfahrungen zwischen den beteiligten Teams und der gesamten Organisation.

Die Praxisgemeinschaft kann vom zentralisierten Praxisteam für Chaos Engineering betrieben werden, aber jeder innerhalb der Organisation kann Mitglied der Community werden. Das zentralisierte Team kann die Community of Practice nutzen, um Neuigkeiten zu verbreiten und Erkenntnisse zu sammeln und Feedback von Teams einzuholen, die die Standards und Verfahren anwenden, die vom zentralisierten Team verwaltet werden. Die Community wird als Feedback-Schleife dienen, um das zentralisierte Team über die Wirksamkeit der Chaos-Engineering-Praktiken in den beteiligten Teams zu informieren. Das zentralisierte Praxisteam kann dann seine Dokumentation und die unterstützenden Artefakte anpassen, um die Produktteams bestmöglich zu unterstützen.

Integration von Chaos Engineering in Ihre betriebliche Resilienz

Ein Chaosexperiment ist eine Investition Ihres Unternehmens, um Vorfälle in der Produktion zu verhindern. Es muss ermittelt werden, wo das Unternehmen die größte Rendite aus dieser Investition erzielen kann. Die Organisation kann mit dem zentralisierten Team für Chaos Engineering zusammenarbeiten, um ihre Standards zu aktualisieren und festzustellen, welche Produkte kritisch genug sind, um Chaosexperimente erforderlich zu machen.

Prozess der Systementwicklung

Chaos-Engineering und Chaos-Experimente sollten im Rahmen des Lebenszyklus einer Anwendung wiederholt durchgeführt werden. Ähnlich wie Teams regelmäßig Notfallwiederherstellungstests durchführen, sollten sie das ganze Jahr über kontinuierlich und regelmäßig Chaos-Experimente und Spieltage durchführen. Dieser Ansatz verbessert die Art und Weise, wie ein Unternehmen Vorfälle antizipiert, beobachtet und darauf reagiert.

Schlussfolgerung

Die Disziplin der Chaostechnik hat in den letzten zehn Jahren einen langen Weg zurückgelegt. Es wurde in einer Vielzahl von Branchen eingesetzt und hat Unternehmen dabei geholfen, belastbare Dienstleistungen zu entwickeln und die Kundenzufriedenheit zu steigern. (Beispiele dafür, wie Unternehmen diese Praktiken umgesetzt haben, finden Sie in Chaos Engineering Stories.) Chaos Engineering ermöglicht es Unternehmen, Risiken für ihre geschäftskritischen Anwendungen zu reduzieren, indem kontrollierte Fehler auf allen Ebenen ihres Anwendungsstapels, einschließlich der Dienste von Cloud-Anbietern, implementiert werden. Die Fähigkeit, einen gesamten Anwendungsstapel kontrolliert zu beeinflussen, ermöglicht eine kontinuierliche Resilienz, Verbesserung der betrieblichen Exzellenz, Beobachtbarkeit und eine wiederherstellungsorientierte Architektur ohne den Stress von Produktionsausfällen. Dieser Ansatz führt auch zu besseren Verfahren für Resilienztests im gesamten Unternehmen. Um mit Chaos Engineering anzufangen, veranstalten Sie einen Chaos-Spieletag oder einen Workshop, um den Wert zu demonstrieren, den Chaos Engineering für Ihr Unternehmen bieten kann.

Ressourcen

AWS FIS Implementierungen von Fehlermodellen:

- Wird verwendet AWS Fault Injection Service , um die Widerstandsfähigkeit von Anwendungen in mehreren Regionen und mehreren AZ zu demonstrieren (Blogbeitrag)AWS
- AWS Der Fault Injection Simulator unterstützt Chaos-Engineering-Experimente auf Amazon EKS-Pods (AWS Blogbeitrag)
- Ankündigung neuer Funktionen des AWS Fault Injection Simulators für Amazon ECS-Workloads (AWS Blogbeitrag)
- Verbessern Sie die Ausfallsicherheit von Anwendungen mit Amazon EBS-Volumenmetriken und dem AWS Fault Injection Simulator (AWS Blogbeitrag)
- Chaos Engineering nutzt den AWS Fault Injection Simulator in einer AWS Umgebung mit mehreren Konten (Blogbeitrag)AWS
- Chaos-Experimente auf Amazon RDS mit dem AWS Fault Injection Simulator (AWS Blogbeitrag)
- Aufbau robuster serverloser Anwendungen mithilfe von Chaos Engineering (AWS Blogbeitrag)
- Verwenden Sie FIS, um eine Spot-Instance zu unterbrechen (Workshop)AWS
- Automatisieren Sie Chaos Engineering in Ihren Liefer-Pipelines (Community-Beitrag)AWS

Andere Ressourcen:

- Investitionen in Chaos Engineering als strategische Notwendigkeit
- Geschichten aus der Chaos-Technik
- CloudWatchAmazon-Dokumentation
- Amazon CloudWatch RUM-Dokumentation
- AWS X-Ray -Dokumentation
- AWS FIS -Dokumentation

Anhang: Musterdokumente

Die in diesem Abschnitt bereitgestellten Beispieldokumente verwenden eine fiktive Website zur Adoption von Haustieren (PetSite) als Zielanwendung für ein Chaosexperiment.

- Dokument zur Planung des Experiments
- Dokument mit den Ergebnissen des Experiments

Dokument zur Planung des Experiments

Stabiler Zustand

Prozessname	Website zur Adoption von Haustieren
Physikalische Architektur	(Link zum Architekturdiagramm.)
Logische Architektur	(Link zum logischen Diagramm.)
Definieren Sie den stationären Zustand	Die durchschnittliche Seitenladezeit, gemessen anhand von Largest Contentful Paint (LCP), beträgt für die Website zur Aufnahme von Haustieren 2,5 Sekunden oder weniger bei einer Latenz von 99 Prozent (P99) von 4,0 Sekunden oder weniger bei einer Ausgangsrate von 5000 gleichzeitigen Benutzern.
Steady-State-Kennzahlen	LCP-Metriken, die für die gesamte Benutzerb asis erfasst wurden, und goldene Metriken (Latenz, Durchsatz, Fehlerraten, Sättigung).
Beobachtbarkeit im stationären Zustand	LCP wird vom Browser des Benutzers erfasst, an Amazon CloudWatch gesendet und mit CloudWatch RUM überprüft. Über einen Zeitraum von 60 Sekunden werden die durchschnittliche und die P99-LCP-Zeit für alle Anfragen in diesem Zeitraum aggregiert. Top-

Prozessname	Website zur Adoption von Haustieren
	level Goldene Metriken werden mithilfe von erfasst. CloudWatch
Prozess zur Erreichung eines stabilen Zustands	Grafana K6 wird verwendet, um eine Last zu erzeugen, die den normalen Produktio nsverkehr von etwa 5.000 gleichzeitigen Benutzern simuliert.

Anforderungen an die Beobachtbarkeit

Teams sollten in der Lage sein, Folgendes zu sehen:

- Steady State: Was wird beobachtet, um sicherzustellen, dass sich die Anwendung unter normalen Bedingungen befindet?
- Fehlerzustand: Wie wird der Fehlerzustand im Dashboard angezeigt? Beispiel:
  - Alarme, die ausgelöst werden sollten
  - Protokolle, die generiert werden sollten
- Auswirkung eines Fehlers: Was sollte beachtet werden, um Komponenten zu erkennen, die voraussichtlich betroffen sein werden (Umfang der Auswirkungen)?
- Wiederherstellung: Wie wird die Wiederherstellung betrachtet und gemessen, um die MTTR zu ermitteln?
- Debug: Einzelheiten zur Fehlerbehebung bei fehlgeschlagenen Experimenten.

Die folgende Tabelle enthält Vorschläge und Beispiele für ein Diagramm mit den Anforderungen an die Beobachtbarkeit. Sie sollten anhand Ihres spezifischen Experiments definieren, was beobachtet werden soll.

Was muss beachtet werden	Link zum Beobachtbarkeitstool	Was wird beobachtet
Quelle der Eingabe	Grafana K6 Armaturenbrett	• Anzahl laufender Container • Anforderungen pro Sekunde

Was muss beachtet werden	Link zum Beobachtbarkeitstool	Was wird beobachtet
Allgemeiner Zustand der Anwendung	• CloudWatch Dashboard zur Adoption von Haustieren • Benutzererfahrungs-Dashboard zur Akzeptanz von Haustieren (RUM)	• Anzahl intakter Amazon EKS-Knoten • CPU-Auslastung des Amazon EKS-Knotens
Integrität des Workflows	CloudWatch Dashboard zur Adoption von Haustieren	LCP-Zeit, goldene Kennzahlen
Ablaufverfolgungen	Dashboard zur Adoption X-Ray von Haustieren	• Latenz anfragen • Anzahl der Anfragen • Anzahl der Fehler
Protokolle	CloudWatch Protokolle zur Adoption von Haustieren	Alle Fehler, auf die die Pods stoßen, werden in CloudWatch Logs gespeichert.

Definition des Experiments

Name des Experiments	CPU-Auslastung des Amazon PetSite EKS-Pods
Experimentieren Sie mit dem Quellcode	(Link zum Quell-Repository für Experimente.)
Beschreibung des Experiments	In diesem Experiment wird untersucht, wie sich eine Erhöhung der CPU-Auslastung des PetSite Anwendungs-Pods auf das allgemeine Kundenerlebnis auswirken würde. Indem wir jedem laufenden PetSite Pod CPU-Stress zuweisen, können wir nachvollziehen, ob

Name des Experiments	CPU-Auslastung des Amazon PetSite EKS-Pods
	es Auswirkungen auf die Kunden gibt und in welchem Ausmaß diese Auswirkungen sind.
Anforderungen oder Parameter des Experiments	Anwendungslast: Produktionsdurchschnitt Auswahl der Pod-Bezeichnung: `app=petsite`
Experimentdauer	10 Minuten
Umgebung	Alpha-Testumgebung
Experimentieren Sie mit Zielressourcen	PetSite Anwendungs-Pods
Ausgangsbasis für das Experiment, die mit dem Tool zur Lastgenerierung eingeführt wird	• 54% der Anfragen haben einen LCP von <2,5 Sekunden. • 46% der Anfragen haben einen LCP von <4 Sekunden. • Es wurden keine Fehler beobachtet.
Backoff-Zustand	Keine

Hypothese

Was wäre wenn	Auswirkung	Wiederherstellung
Was würde mit dem Steady State passieren, wenn die PetSite Anwendungs-Pods bei normalem Datenverkehr auf Produktionsebene 10 Minuten lang eine CPU-Auslastung von	Die LCP-Zeiten bleiben für P50 der Benutzer unter 2,5 Sekunden, während P99 bei 4,0 Sekunden oder weniger liegt. Der Verbraucher sollte in der Lage sein, die Landingpage zu laden. PetSite	Erkennung: • CPU-Auslastung wird durch Alarme erkannt, die in konfiguriert sind CloudWatch.

Was wäre wenn	Auswirkung	Wiederherstellung
mehr als 60% erfahren oder verursachen würden?		• LCP-Metriken werden auch Alarme für die Verschlec hterung der Benutzere rfahrung auslösen. Self-healing: • Der verteilte Charakter der Microservice-Archi tektur bedeutet, dass viele Instanzen von Pods in mehreren Availability Zones ausgeführt werden. • Die EKS-Cluster-Steuer ebene leitet den Datenverk ehr von den betroffenen Pods weg und startet neue Pods auf Worker-Knoten. Wiederherstellung: Wenn die CPU-Auslastung wieder normal ist, sollte das LCP automatisch wiederher gestellt werden.

Ablauf des Experiments

Passen Sie das folgende Beispiel Schritt für Schritt an Ihr spezifisches Experiment an:

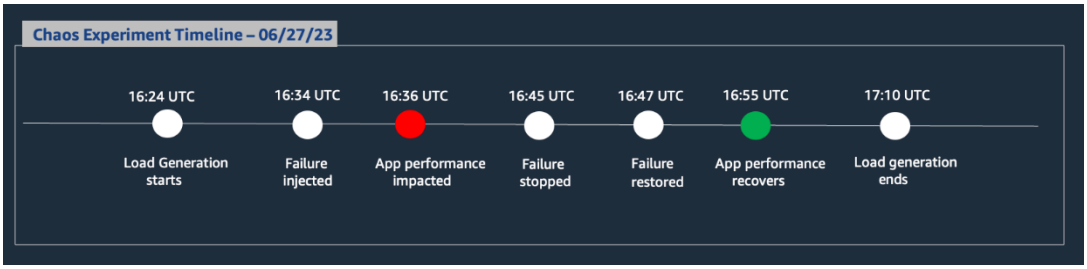
1. Überprüfen Sie den Zugriff auf und die Funktionalität aller Amazon- CloudWatch, CloudWatch RUM- und AWS X-Ray Dashboards.
2. Überprüfen Sie den Zustand der Anwendungsumgebung:

a. Vergewissern Sie sich mithilfe des CloudWatch Dashboards, dass der EKS-Cluster fehlerfrei ist.
b. Rufen Sie unter der Beispiel-URL die Anwendungsbereitstellung auf der Testseite zur Adoption von Haustieren auf.

3. Starten Sie einen Ladevorgang, um einen stabilen Zustand zu erreichen:
   a. Vergewissern Sie sich, dass der Lastgenerator läuft und 5000 Anfragen pro Sekunde sendet.
   b. Warten Sie 5 Minuten, bis die Anwendung ihren stationären Status erreicht hat.
   c. Bestätigen Sie den stabilen Status der Anwendung mithilfe des CloudWatch RUM-Dashboards.
4. Einen Fehler einleiten (Experiment):
   a. Öffnen Sie die AWS FIS Konsole.
   b. Führen Sie das Experiment zur Adoption von Haustieren mit Pod-Stress durch.
   c. Vergewissern Sie sich, dass das Experiment in der Konsole ausgeführt wird.
5. Beobachten Sie die Auswirkungen des Fehlers auf Ihre Anwendung:
   a. Erfassen Sie Screenshots aus dem CloudWatch RUM und den CloudWatch Dashboards und notieren Sie sich alle anomalen Datenpunkte.
   b. Machen Sie nach Abschluss des Experiments zusätzliche Screenshots AWS FIS, um aufzuzeichnen, ob die Anwendung ohne Stress wieder in den stationären Zustand zurückkehrt, und notieren Sie alle Anomalien in den Datenpunkten.
   c. Wenn der stationäre Zustand nicht wieder aufgenommen wird, ergreifen Sie Maßnahmen, um die Anwendung wiederherzustellen, und zeichnen Sie die unternommenen Schritte auf.
6. Stellen Sie sicher, dass die Umgebung wieder normal ist:
   - Überprüfen Sie alle Kennzahlen zu Geschäft, Benutzererfahrung, Anwendung und Infrastruktur, um sicherzustellen, dass das System wieder in einen bekannten Zustand zurückgekehrt ist. Erfassen Sie Dashboard-Screenshots, falls hilfreich.

Zeitplan für das Experiment

Stellen Sie sicher, dass Sie den Zeitplan für das gesamte Experiment festhalten, angefangen bei der Lastgenerierung, der Fehlereinschleusung, der Beobachtung der Auswirkungen und der Wiederherstellung der Anwendung bis hin zum Stopp der Lastgenerierung. Dies wird im folgenden Beispiel veranschaulicht.



Ergebnisse des Experiments

ID des Versuchslaufs	Ergebnisse des Experiments
PET-ADOPT-EXP-23	(Link zu den Versuchsergebnissen.)

Identifizierte Mängel

- Der Kubernetes-Cluster hat die CPU-Beeinträchtigung der PetSite Pods nicht erkannt und daher keine zusätzlichen Bereitstellungen geplant.
- Infolge dieses Experiments gab es keinen Anstieg der 4XX- oder 5XX-Fehlerraten.
- Wir müssen den Integritätscheck des Pods anpassen, um die Auswirkungen auf LCP zu berücksichtigen, wenn es zu Ressourcenengpässen kommt.

Dokument mit den Ergebnissen des Experiments

Konfiguration

Dokumentieren Sie die spezifischen Konfigurationen für das Experiment. Beispiel:

- Das System zur Lastgenerierung simuliert 5.000 Benutzer, die insgesamt 85 Anfragen pro Sekunde ausgeben.

Voraussetzungen

- Es wurde überprüft, ob die Website zur Adoption von Haustieren in der Alphatestumgebung lief.

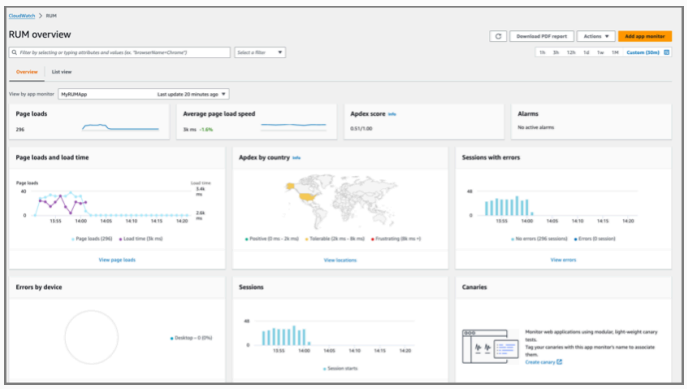
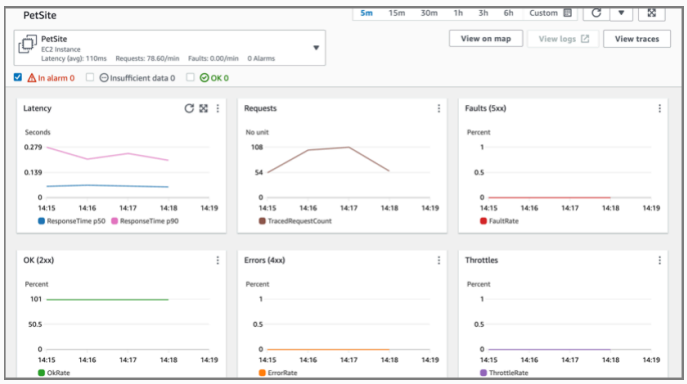
- Es wurde überprüft, ob die Versuchsvorlage so konfiguriert wurde, dass die PetSite Anwendungs-Pods, die im EKS-Cluster ausgeführt werden, mit CPU-Belastung stress werden. Anwendungs-Pods wurden anhand des Kubernetes-Labels identifiziert. `app=petsite`
- Es wurde bestätigt, dass Load läuft und 85 Anfragen pro Sekunde generiert.

Stetiger Zustand

Dokumentieren Sie die Schritte, die Sie unternommen haben, um den Steady-State zu erreichen, und wie Sie ihn verifiziert haben. Beispiel:

Für den Testeinsatz an der Adoptionsstelle für Haustiere wird eine Last von 85 RPS erzeugt, um den stationären Zustand zu simulieren. Vor der Durchführung des CloudWatch Experiments wurden CloudWatch RUM und Dashboards überprüft, um sicherzustellen, dass alle Geschäfts- und Anwendungskennzahlen im normalen Bereich lagen.

Daten zur Beobachtbarkeit:

Expected	Beobachtet
• LCP dauert weniger als 4 Sekunden für P99 der Anfragen. • Die Antwortlatenz beträgt weniger als 500 ms. • Es liegen keine 4XX- oder 5XX-Fehler vor.	 

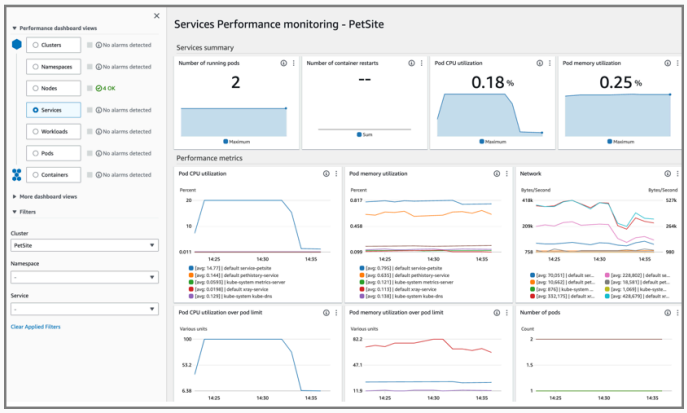
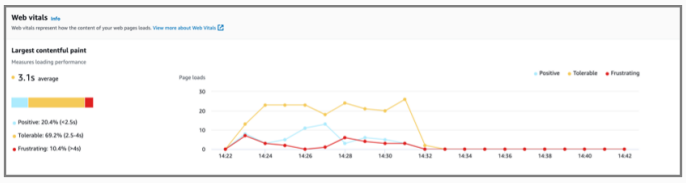
Fehlerinjektion

AWS FIS wurde verwendet, um Fehler mithilfe der Versuchsvorlage zu injizieren (Link angeben). Das Experiment sollte 10 Minuten dauern, und es wurde ein Rollback konfiguriert, wenn die Workerknoten einer CPU-Belastung von über 60 Prozent stress waren.

Beobachtung des Fehlers

Der CloudWatch RUM und die CloudWatch Dashboards wurden überprüft, um den stabilen Zustand der Anwendung nachzuverfolgen (definiert anhand von LCP-Metriken). Screenshots wurden in der folgenden Tabelle erfasst.

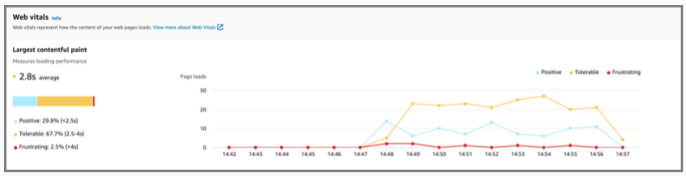
Daten zur Beobachtbarkeit:

Expected	Beobachtet
• LCP sollte für P99 unter 4 Sekunden bleiben. • Die Reaktionszeit sollte unter 500 ms bleiben. • Es sollten keine 4XX- oder 5XX-Fehler auftreten.	 

Wiederherstellung

Nachdem der Stress entfernt wurde (das AWS FIS Experiment ist abgeschlossen und die CPU-Belastung der Pods wurde entfernt), sollte die Anwendung ihren normalen stationären Zustand wieder aufnehmen. Es sollte kein manuelles Eingreifen erforderlich sein.

Daten zur Beobachtbarkeit:

Expected	Beobachtet (Bildschirmfoto)
LCP P99 sollte unter 4 Sekunden liegen, der Durchschnitt unter 2,5 Sekunden.	

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Erste Veröffentlichung	—	4. April 2025

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.