Rahmen für den Lebenszyklus der Widerstandsfähigkeit

AWS Präskriptive Leitlinien

AWS Präskriptive Leitlinien: Rahmen für den Lebenszyklus der Widerstandsfähigkeit

Table of Contents

Rahmen für den Lebenszyklus von Resilienz: Ein kontinuierlicher Ansatz zur Verbesserung der Widerstandsfähigkeit

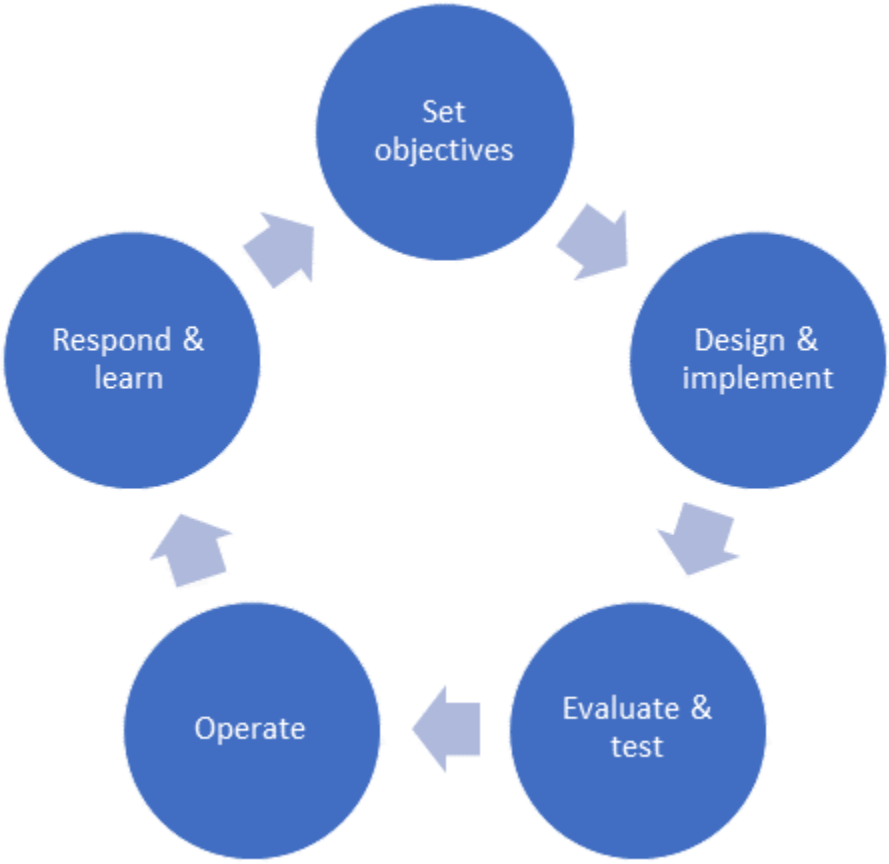
Amazon Web Services (Mitwirkende)

Oktober 2023 (Verlauf der Dokumente)

Moderne Unternehmen sehen sich heute mit einer ständig wachsenden Anzahl von Herausforderungen im Zusammenhang mit ihrer Resilienz konfrontiert, vor allem, weil sich die Erwartungen der Kunden hin zu einer „Alway-On“ -Mentalität und Verfügbarkeit verlagern. Remote-Teams und komplexe, verteilte Anwendungen gehen mit einem steigenden Bedarf an häufigen Releases einher. Infolgedessen müssen ein Unternehmen und seine Anwendungen widerstandsfähiger denn je sein.

AWS definiert Resilienz als die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich davon zu erholen, einschließlich solcher, die mit der Infrastruktur, abhängigen Diensten, Fehlkonfigurationen und vorübergehenden Netzwerkproblemen zusammenhängen. (Weitere Informationen finden Sie unter Resilienz und Zuverlässigkeitskomponenten in der Dokumentation Reliability Pillar des AWS Well-Architected Framework.) Um das gewünschte Maß an Resilienz zu erreichen, sind jedoch häufig Kompromisse erforderlich. Die betriebliche Komplexität, die technische Komplexität und die Kosten müssen bewertet und entsprechend angepasst werden.

Auf der Grundlage jahrelanger Zusammenarbeit mit Kunden und internen Teams AWS hat das Unternehmen ein Resilienz-Lifecycle-Framework entwickelt, das Erkenntnisse und bewährte Verfahren im Bereich Resilienz erfasst. Das Framework umreißt fünf wichtige Phasen, die in der folgenden Abbildung dargestellt werden. In jeder Phase können Sie Strategien, Dienste und Mechanismen nutzen, um Ihre Resilienz zu verbessern.



Diese Phasen werden in den folgenden Abschnitten dieses Leitfadens erörtert:

- Phase 1: Ziele setzen
- Phase 2: Konzipieren und Implementieren
- Phase 3: Evaluieren und testen
- Stufe 4: Operieren
- Phase 5: Reagieren und lernen

Begriffe und Definitionen

Die Resilienzkonzepte jeder Phase werden auf unterschiedlichen Ebenen angewendet und reichen von einzelnen Komponenten bis hin zu ganzen Systemen. Die Umsetzung dieser Konzepte erfordert eine klare Definition mehrerer Begriffe:

- Eine Komponente ist ein Element, das eine Funktion erfüllt und aus Software- und Technologieressourcen besteht. Zu den Komponenten gehören beispielsweise Codekonfiguration,

Infrastruktur wie Netzwerke oder sogar Server, Datenspeicher und externe Abhängigkeiten wie Geräte mit Multi-Faktor-Authentifizierung (MFA).

- Eine Anwendung ist eine Sammlung von Komponenten, die einen geschäftlichen Nutzen bieten, wie z. B. ein kundenorientiertes Web-Storefront oder der Backend-Prozess, der Modelle für maschinelles Lernen verbessert. Eine Anwendung kann aus einer Teilmenge von Komponenten in einem einzigen AWS Konto bestehen, oder es kann sich um eine Sammlung mehrerer Komponenten handeln, die sich über mehrere Regionen erstrecken. AWS-Konten
- Ein System ist eine Sammlung von Anwendungen, Personen und Prozessen, die zur Verwaltung einer bestimmten Geschäftsfunktion erforderlich sind. Es umfasst die Anwendung, die für die Ausführung einer Funktion erforderlich ist, betriebliche Prozesse wie Continuous Integration and Continuous Delivery (CI/CD), Observability, Konfigurationsmanagement, Reaktion auf Vorfälle und Disaster Recovery sowie die Bediener, die diese Aufgaben verwalten.
- Eine Störung ist ein Ereignis, das Ihre Anwendung daran hindert, ihre Geschäftsfunktionen ordnungsgemäß zu erfüllen.
- Beeinträchtigung ist die Auswirkung, die eine Störung auf eine Anwendung hat, wenn sie nicht gemildert wird. Anwendungen können beeinträchtigt werden, wenn sie einer Reihe von Störungen ausgesetzt sind.

Kontinuierliche Belastbarkeit

Der Resilienz-Lebenszyklus ist ein fortlaufender Prozess. Selbst innerhalb derselben Organisation können Ihre Anwendungsteams je nach den Anforderungen Ihrer Anwendung in jeder Phase unterschiedliche Leistungen erbringen. Je vollständiger jedoch jede Phase ist, desto robuster ist Ihre Anwendung.

Sie sollten sich den Resilienz-Lebenszyklus als einen Standardprozess vorstellen, den Ihr Unternehmen operationalisieren kann. AWS hat den Resilienz-Lebenszyklus bewusst so modelliert, dass er dem Software Development Lifecycle (SDLC) ähnelt, mit dem Ziel, Planung, Tests und Lernen in alle Betriebsprozesse einzubeziehen, während Sie Ihre Anwendungen entwickeln und betreiben. Wie bei vielen agilen Entwicklungsprozessen kann der Resilienz-Lebenszyklus bei jeder Iteration des Entwicklungsprozesses wiederholt werden. Wir empfehlen Ihnen, die Praktiken in jeder Phase des Lebenszyklus im Laufe der Zeit schrittweise zu vertiefen.

Phase 1: Ziele setzen

Die Grundlage für die Phase der gesetzten Ziele ist es, zu verstehen, welches Maß an Resilienz erforderlich ist und wie Sie es messen werden. Es ist schwierig, etwas zu verbessern, wenn Sie kein Ziel haben und es nicht messen können.

Nicht alle Anwendungen benötigen das gleiche Maß an Belastbarkeit. Wenn Sie sich Ziele setzen, sollten Sie das erforderliche Niveau berücksichtigen, um die richtigen Investitionen und Kompromisse zu tätigen. Eine gute Analogie hierfür ist ein Auto: Es hat vier Reifen, trägt aber nur ein Reserverad. Die Wahrscheinlichkeit, während einer Fahrt mehrere platten Reifen zu bekommen, ist gering, und zusätzliche Ersatzteile könnten andere Merkmale wie den Laderaum oder die Kraftstoffeffizienz beeinträchtigen, sodass dies ein vernünftiger Kompromiss ist.

Nachdem Sie die Ziele definiert haben, implementieren Sie in späteren Phasen (Phase 2: Planung und Implementierung und Phase 4: Betrieb) Kontrollen der Beobachtbarkeit, um zu ermitteln, ob die Ziele erreicht werden.

Zuordnung kritischer Anwendungen

Die Definition von Resilienzzielen sollte nicht ausschließlich ein technisches Gespräch sein. Beginnen Sie stattdessen mit einem geschäftsorientierten Fokus, um zu verstehen, was die Anwendung bieten sollte und welche Folgen eine Beeinträchtigung hat. Dieses Verständnis der Geschäftsziele überträgt sich dann auf Bereiche wie Architektur, Technik und Betrieb. Alle von Ihnen definierten Stabilitätsziele können auf alle Ihre Anwendungen angewendet werden, aber die Art und Weise, wie die Ziele gemessen werden, hängt häufig von der Funktion der Anwendung ab. Möglicherweise führen Sie eine Anwendung aus, die für Ihr Unternehmen von entscheidender Bedeutung ist, und wenn diese Anwendung beeinträchtigt wird, könnte Ihr Unternehmen erhebliche Umsatzeinbußen oder Reputationsschäden erleiden. Alternativ haben Sie möglicherweise eine andere Anwendung, die nicht so wichtig ist und einige Ausfallzeiten tolerieren kann, ohne die Geschäftsfähigkeit Ihres Unternehmens zu beeinträchtigen.

Stellen Sie sich als Beispiel eine Auftragsverwaltungsanwendung für ein Einzelhandelsunternehmen vor. Wenn die Komponenten der Auftragsverwaltungsanwendung beeinträchtigt sind und nicht ordnungsgemäß funktionieren, werden neue Verkäufe nicht abgeschlossen. Dieses Einzelhandelsunternehmen hat auch ein Café für seine Mitarbeiter, das sich in einem seiner Gebäude befindet. Das Café verfügt über ein Online-Menü, auf das die Mitarbeiter auf einer statischen Webseite zugreifen können. Wenn diese Webseite nicht mehr verfügbar ist, können sich einige

Mitarbeiter beschweren, aber das wird dem Unternehmen nicht unbedingt finanziellen Schaden zufügen. Auf der Grundlage dieses Beispiels würde sich das Unternehmen wahrscheinlich für aggressivere Stabilitätsziele für die Auftragsverwaltungsanwendung entscheiden, aber keine nennenswerten Investitionen tätigen, um die Ausfallsicherheit der Webanwendung sicherzustellen.

Die Identifizierung der kritischsten Anwendungen, der Bereiche, in denen der größte Aufwand erforderlich ist und wo Kompromisse eingegangen werden müssen, ist genauso wichtig wie die Fähigkeit, die Widerstandsfähigkeit einer Anwendung in der Produktion zu messen. Um die Auswirkungen einer Wertminderung besser zu verstehen, können Sie eine Business Impact Analysis (BIA) durchführen. Eine BIA bietet einen strukturierten und systematischen Ansatz zur Identifizierung und Priorisierung kritischer Geschäftsanwendungen, zur Bewertung potenzieller Risiken und Auswirkungen sowie zur Identifizierung unterstützender Abhängigkeiten. Die BIA hilft dabei, die Kosten von Ausfallzeiten für die wichtigsten Anwendungen Ihres Unternehmens zu quantifizieren. Anhand dieser Kennzahl lässt sich abschätzen, wie viel es kosten wird, wenn eine bestimmte Anwendung beeinträchtigt ist und ihre Funktion nicht erfüllen kann. Im vorherigen Beispiel könnte das Einzelhandelsgeschäft erhebliche Umsatzeinbußen hinnehmen, wenn die Anwendung zur Auftragsverwaltung beeinträchtigt ist.

Zuordnung von Benutzergeschichten

Während des BIA-Prozesses stellen Sie möglicherweise fest, dass eine Anwendung für mehr als eine Geschäftsfunktion verantwortlich ist oder dass eine Geschäftsfunktion mehrere Anwendungen erfordert. Im vorherigen Beispiel eines Einzelhandelsunternehmens sind für die Auftragsverwaltungsfunktion möglicherweise separate Anwendungen für Checkout, Werbung und Preisgestaltung erforderlich. Wenn eine Anwendung ausfällt, können sich dies sowohl auf das Unternehmen als auch auf die Benutzer auswirken, die mit dem Unternehmen interagieren. Beispielsweise ist das Unternehmen möglicherweise nicht in der Lage, neue Bestellungen hinzuzufügen, Zugang zu Werbeaktionen und Rabatten zu gewähren oder den Preis seiner Produkte zu aktualisieren. Diese verschiedenen Funktionen, die für die Auftragsverwaltungsfunktion erforderlich sind, können auf mehreren Anwendungen basieren. Diese Funktionen können auch mehrere externe Abhängigkeiten aufweisen, was den Prozess der Erzielung einer ausschließlich komponentenorientierten Resilienz zu komplex macht. Eine bessere Möglichkeit, mit diesem Szenario umzugehen, besteht darin, sich auf Benutzerberichte zu konzentrieren, in denen beschrieben wird, welche Erfahrung Benutzer bei der Interaktion mit einer oder mehreren Anwendungen erwarten.

Wenn Sie sich auf Anwenderberichte konzentrieren, können Sie besser verstehen, welche Aspekte des Kundenerlebnisses am wichtigsten sind, sodass Sie Mechanismen zum Schutz vor bestimmten

Bedrohungen entwickeln können. Im vorherigen Beispiel könnte es sich bei einer User Story um Checkout handeln, die die Checkout-Anwendung beinhaltet und von der Preisgestaltungsanwendung abhängig ist. Eine andere User Story könnte die Anzeige von Werbeaktionen sein, was auch die Werbeanwendung beinhaltet. Nachdem Sie die wichtigsten Anwendungen und ihre User Stories zugeordnet haben, können Sie damit beginnen, die Metriken zu definieren, anhand derer Sie die Widerstandsfähigkeit dieser User Stories messen werden. Diese Kennzahlen können auf ein gesamtes Portfolio oder auf einzelne User Stories angewendet werden.

Definition von Messungen

Recovery Point Objectives (RPOs), Recovery Time Objectives (RTOs) und Service Level Objectives (SLOs) sind branchenübliche Messgrößen, mit denen die Widerstandsfähigkeit eines bestimmten Systems bewertet wird. RPO bezieht sich darauf, wie viel Datenverlust das Unternehmen bei einem Ausfall tolerieren kann, wohingegen RTO ein Maß dafür ist, wie schnell eine Anwendung nach einem Ausfall wieder verfügbar sein muss. Diese beiden Kennzahlen werden in Zeiteinheiten gemessen: Sekunden, Minuten und Stunden. Sie können auch messen, wie lange die Anwendung ordnungsgemäß funktioniert, d. h. sie erfüllt ihre Funktionen wie vorgesehen und ist für ihre Benutzer zugänglich. Diese SLOs beschreiben das erwartete Serviceniveau, das Kunden erhalten werden, und werden anhand von Kennzahlen wie dem Prozentsatz (%) der Anfragen gemessen, die innerhalb einer Antwortzeit von weniger als einer Sekunde ohne Fehler bearbeitet werden (z. B. erhalten 99,99% der Anfragen jeden Monat eine Antwort). RPO und RTO beziehen sich auf Disaster-Recovery-Strategien, wobei davon ausgegangen wird, dass es zu Unterbrechungen des Anwendungsbetriebs und der Wiederherstellungsprozesse kommen wird, die von der Wiederherstellung von Backups bis hin zur Umleitung von Benutzerdatenverkehr reichen. SLOs werden durch die Implementierung von Hochverfügbarkeitskontrollen behoben, die dazu neigen, die Ausfallzeiten einer Anwendung zu reduzieren.

SLO-Metriken werden häufig bei der Definition von Service Level Agreements (SLAs) verwendet, bei denen es sich um Verträge zwischen Dienstanbietern und Endbenutzern handelt. SLAs sind in der Regel mit finanziellen Verpflichtungen verbunden und enthalten Strafen, die vom Anbieter zu zahlen sind, wenn diese Vereinbarungen nicht eingehalten werden. Eine SLA ist jedoch kein Maßstab für Ihre Resilienz, und eine Erhöhung einer SLA macht Ihre Anwendung nicht widerstandsfähiger.

Sie können damit beginnen, Ihre Ziele auf der Grundlage von SLOs, RPOs und RTOs festzulegen. Nachdem Sie Ihre Resilienzziele definiert und sich ein klares Bild von Ihren RPO- und RTO-Zielen gemacht haben, können Sie eine Bewertung Ihrer Architektur durchführen, AWS Resilience Hubum potenzielle Schwächen im Zusammenhang mit der Resilienz aufzudecken. AWS Resilience Hub

bewertet eine Anwendungsarchitektur anhand der Best Practices von AWS Well-Architected Framework und gibt Empfehlungen zur Problembehebung im Kontext dessen, was speziell verbessert werden muss, um Ihre definierten RTO- und RPO-Ziele zu erreichen.

Erstellung zusätzlicher Messungen

RPO, RTO und SLOs sind gute Indikatoren für Resilienz, aber Sie können Ziele auch aus geschäftlicher Sicht betrachten und Ziele rund um die Funktionen Ihrer Anwendung definieren. Ihr Ziel könnte beispielsweise lauten: Erfolgreiche Bestellungen pro Minute bleiben über 98%, wenn die Latenz zwischen meinem Frontend und meinem Backend um 40% steigt.Oder: Pro Sekunde gestartete Streams bleiben innerhalb einer Standardabweichung vom Durchschnitt, auch wenn eine bestimmte Komponente verloren geht. Sie können auch Ziele festlegen, um die mittlere Wiederherstellungszeit (MTTR) bei bekannten Fehlertypen zu reduzieren. Beispiel: Die Wiederherstellungszeiten werden um x% reduziert, wenn eines dieser bekannten Probleme auftritt. Durch die Festlegung von Zielen, die sich an den Geschäftsanforderungen orientieren, können Sie die Arten von Ausfällen vorhersehen, die Ihre Anwendung tolerieren sollte. Es hilft Ihnen auch dabei, Ansätze zu finden, mit denen Sie die Wahrscheinlichkeit einer Beeinträchtigung Ihrer Anwendung verringern können.

Wenn Sie über das Ziel nachdenken, den Betrieb fortzusetzen, wenn Sie 5% der Instances verlieren, die Ihre Anwendung unterstützen, könnten Sie entscheiden, dass Ihre Anwendung vorab skaliert werden sollte oder die Möglichkeit haben sollte, schnell genug zu skalieren, um den zusätzlichen Datenverkehr zu bewältigen, der bei diesem Ereignis entsteht. Oder Sie könnten entscheiden, dass Sie verschiedene Architekturmuster nutzen sollten, wie im Abschnitt Phase 2: Entwerfen und Implementieren beschrieben.

Sie sollten auch Messgrößen zur Beobachtung Ihrer spezifischen Geschäftsziele implementieren. Sie können beispielsweise die durchschnittliche Bestellrate, den durchschnittlichen Bestellpreis, die durchschnittliche Anzahl von Abonnements oder andere Kennzahlen verfolgen, die anhand des Verhaltens Ihrer Anwendung Aufschluss über den Zustand Ihres Unternehmens geben können. Durch die Implementierung von Beobachtungsfunktionen für Ihre Anwendung können Sie Alarme einrichten und Maßnahmen ergreifen, wenn diese Messwerte Ihre definierten Grenzen überschreiten. Die Beobachtbarkeit wird im Abschnitt Phase 4: Betrieb ausführlicher behandelt.

Phase 2: Design und Implementierung

In der vorherigen Phase legen Sie Ihre Resilienzziele fest. In der Entwurfs- und Implementierungsphase versuchen Sie nun, Ausfallursachen zu antizipieren und Entwurfsoptionen zu ermitteln, wobei Sie sich an den Zielen orientieren, die Sie sich in der vorherigen Phase gesetzt haben. Außerdem definieren Sie Strategien für das Änderungsmanagement und entwickeln Softwarecode und die Infrastrukturkonfiguration. In den folgenden Abschnitten werden AWS bewährte Methoden beschrieben, die Sie berücksichtigen sollten, während Sie Kompromisse wie Kosten, Komplexität und Betriebskosten berücksichtigen sollten.

AWS Well-Architected Framework

Wenn Sie Ihre Anwendung auf der Grundlage Ihrer gewünschten Stabilitätsziele konzipieren, müssen Sie mehrere Faktoren bewerten und Kompromisse bei der optimalen Architektur eingehen. Um eine äußerst robuste Anwendung zu entwickeln, müssen Sie Aspekte wie Design, Aufbau und Bereitstellung, Sicherheit und Betrieb berücksichtigen. Das AWS Well-Architected Framework bietet eine Reihe von Best Practices, Entwurfsprinzipien und Architekturmustern, mit denen Sie robuste Anwendungen entwerfen können. AWS Die sechs Säulen des AWS Well-Architected Framework bieten bewährte Verfahren für die Entwicklung und den Betrieb belastbarer, sicherer, effizienter, kostengünstiger und nachhaltiger Systeme. Das Framework bietet eine Möglichkeit, Ihre Architekturen konsistent anhand bewährter Verfahren zu messen und Verbesserungspotenziale zu identifizieren.

Im Folgenden finden Sie Beispiele dafür, wie das AWS Well-Architected Framework Ihnen helfen kann, Anwendungen zu entwerfen und zu implementieren, die Ihre Resilienzziele erfüllen:

- Die Säule Zuverlässigkeit: Die Säule Zuverlässigkeit betont, wie wichtig es ist, Anwendungen zu entwickeln, die auch bei Ausfällen oder Störungen korrekt und konsistent funktionieren. Das AWS Well-Architected Framework empfiehlt beispielsweise, eine Microservices-Architektur zu verwenden, um Ihre Anwendungen kleiner und einfacher zu gestalten, sodass Sie zwischen den Verfügbarkeitsanforderungen verschiedener Komponenten innerhalb Ihrer Anwendung unterscheiden können. Sie finden dort auch detaillierte Beschreibungen der bewährten Methoden für die Erstellung von Anwendungen mithilfe von Drosselung, Wiederholungsversuchen mit exponentiellem Back-Off, Fail Fast (Load Shedding), Idempotenz, konstanter Arbeit, Schutzschaltern und statischer Stabilität.
- Umfassende Überprüfung: Das AWS Well-Architected Framework fördert eine umfassende Überprüfung Ihrer Architektur anhand von Best Practices und Entwurfsprinzipien. Es bietet

eine Möglichkeit, Ihre Architekturen konsistent zu messen und Verbesserungspotenziale zu identifizieren.

- Risikomanagement: Das AWS Well-Architected Framework hilft Ihnen dabei, Risiken zu identifizieren und zu managen, die sich auf die Zuverlässigkeit Ihrer Anwendung auswirken könnten. Indem Sie potenzielle Ausfallszenarien proaktiv angehen, können Sie deren Wahrscheinlichkeit oder die daraus resultierende Beeinträchtigung verringern.
- Kontinuierliche Verbesserung: Resilienz ist ein fortlaufender Prozess, und das AWS Well-Architected Framework legt Wert auf kontinuierliche Verbesserung. Indem Sie Ihre Architektur und Prozesse auf der Grundlage der Leitlinien des AWS Well-Architected Framework regelmäßig überprüfen und verfeinern, können Sie sicherstellen, dass Ihre Systeme angesichts sich ändernder Herausforderungen und Anforderungen widerstandsfähig bleiben.

Abhängigkeiten verstehen

Das Verständnis der Abhängigkeiten eines Systems ist entscheidend für die Widerstandsfähigkeit. Zu den Abhängigkeiten gehören Verbindungen zwischen Komponenten innerhalb einer Anwendung und Verbindungen zu Komponenten außerhalb der Anwendung, z. B. APIs von Drittanbietern und unternehmenseigenen Shared Services. Wenn Sie diese Verbindungen verstehen, können Sie Störungen isolieren und bewältigen, da sich eine Beeinträchtigung einer Komponente auf andere Komponenten auswirken kann. Dieses Wissen hilft Technikern, die Auswirkungen von Beeinträchtigungen einzuschätzen, entsprechend zu planen und sicherzustellen, dass Ressourcen effektiv genutzt werden. Das Verständnis von Abhängigkeiten hilft Ihnen, alternative Strategien zu entwickeln und Wiederherstellungsprozesse zu koordinieren. Es hilft Ihnen auch dabei, Fälle zu ermitteln, in denen Sie eine harte Abhängigkeit durch eine weiche Abhängigkeit ersetzen können, sodass Ihre Anwendung auch bei einer Beeinträchtigung der Abhängigkeit weiterhin ihre Geschäftsfunktion erfüllen kann. Abhängigkeiten beeinflussen auch Entscheidungen zum Lastenausgleich und zur Anwendungsskalierung. Das Verständnis von Abhängigkeiten ist wichtig, wenn Sie Änderungen an Ihrer Anwendung vornehmen, da es Ihnen helfen kann, potenzielle Risiken und Auswirkungen zu ermitteln. Dieses Wissen hilft Ihnen bei der Entwicklung stabiler, robuster Anwendungen und unterstützt Sie bei der Fehlerverwaltung, Folgenabschätzung, Wiederherstellung von Störungen, Lastenausgleich, Skalierung und Änderungsmanagement. Sie können Abhängigkeiten manuell nachverfolgen oder Tools und Dienste verwenden, AWS X-Rayum beispielsweise die Abhängigkeiten Ihrer verteilten Anwendungen zu verstehen.

Strategien für die Notfallwiederherstellung

Eine Disaster-Recovery-Strategie (DR) spielt eine zentrale Rolle bei der Entwicklung und dem Betrieb robuster Anwendungen, vor allem durch die Sicherstellung der Geschäftskontinuität. Sie garantiert, dass wichtige Geschäftsabläufe auch bei Katastrophenereignissen mit der geringstmöglichen Beeinträchtigung fortgeführt werden können, wodurch Ausfallzeiten und potenzielle Umsatzverluste minimiert werden. DR-Strategien sind für den Datenschutz unverzichtbar, da sie häufig regelmäßige Datensicherungen und Datenreplikation über mehrere Standorte hinweg beinhalten. Dadurch werden wertvolle Geschäftsinformationen geschützt und Totalverluste im Katastrophenfall vermieden. Darüber hinaus unterliegen viele Branchen Richtlinien, nach denen Unternehmen über eine DR-Strategie verfügen müssen, um sensible Daten zu schützen und sicherzustellen, dass die Dienste im Notfall verfügbar bleiben. Durch die Sicherstellung minimaler Beeinträchtigungen der Dienste stärkt eine DR-Strategie auch das Vertrauen und die Zufriedenheit der Kunden. Eine gut umgesetzte und häufig angewandte DR-Strategie reduziert die Wiederherstellungszeit nach einem Notfall und trägt dazu bei, dass Anwendungen schnell wieder online sind. Darüber hinaus können Katastrophen zu erheblichen Kosten führen, nicht nur aufgrund von Umsatzeinbußen aufgrund von Ausfallzeiten, sondern auch aufgrund der Kosten für die Wiederherstellung von Anwendungen und Daten. Eine gut durchdachte DR-Strategie schützt vor diesen finanziellen Verlusten.

Welche Strategie Sie wählen, hängt von den spezifischen Anforderungen Ihrer Anwendung, Ihrem RTO und RPO sowie Ihrem Budget ab. AWS Elastic Disaster Recoveryist ein speziell entwickelter Resilienz-Service, mit dem Sie Ihre DR-Strategie sowohl für lokale als auch für cloudbasierte Anwendungen umsetzen können.

Weitere Informationen finden Sie auf der Website unter Disaster Recovery of Workloads on AWS und AWS Multi-Region Fundamentals. AWS

Definition von CI/CD-Strategien

Eine der häufigsten Ursachen für Beeinträchtigungen von Anwendungen sind Code- oder andere Änderungen, die die Anwendung gegenüber einem zuvor bekannten Betriebszustand verändern. Wenn Sie das Änderungsmanagement nicht sorgfältig angehen, kann es zu häufigen Beeinträchtigungen kommen. Die Häufigkeit von Änderungen erhöht die Möglichkeit, Auswirkungen zu erzielen. Seltenere Änderungen führen jedoch zu größeren Änderungen, bei denen die Wahrscheinlichkeit, dass sie zu Wertminderungen führen, aufgrund ihrer hohen Komplexität viel höher ist. Die Verfahren zur kontinuierlichen Integration und kontinuierlichen Bereitstellung (CI/CD) sind darauf ausgelegt, kleine und häufige Änderungen vorzunehmen (was zu einer höheren

Produktivität führt) und gleichzeitig jede Änderung durch Automatisierung einem hohen Maß an Inspektion zu unterziehen. Einige der grundlegenden Strategien sind:

- Vollständige Automatisierung: Das grundlegende Konzept von CI/CD besteht darin, die Erstellungs- und Bereitstellungsprozesse so weit wie möglich zu automatisieren. Dies umfasst das Erstellen, Testen, Bereitstellen und sogar die Überwachung. Automatisierte Pipelines tragen dazu bei, die Wahrscheinlichkeit menschlicher Fehler zu verringern, Konsistenz zu gewährleisten und den Prozess zuverlässiger und effizienter zu gestalten.
- Testgetriebene Entwicklung (TDD): Schreiben Sie Tests, bevor Sie den Anwendungscode schreiben. Diese Vorgehensweise stellt sicher, dass dem gesamten Code Tests zugeordnet sind, wodurch die Zuverlässigkeit des Codes und die Qualität der automatisierten Inspektion verbessert werden. Diese Tests werden in der CI-Pipeline ausgeführt, um Änderungen zu validieren.
- Häufige Commits und Integrationen: Ermutigen Sie Entwickler, Code häufig zu übertragen und Integrationen häufig durchzuführen. Kleine, häufige Änderungen lassen sich leichter testen und debuggen, wodurch das Risiko schwerwiegender Probleme verringert wird. Durch die Automatisierung werden die Kosten für jeden Commit und jeder Bereitstellung reduziert, sodass häufige Integrationen möglich sind.
- Unveränderliche Infrastruktur: Behandeln Sie Ihre Server und andere Infrastrukturkomponenten wie statische, unveränderliche Entitäten. Ersetzen Sie die Infrastruktur, anstatt sie so weit wie möglich zu modifizieren, und bauen Sie eine neue Infrastruktur mithilfe von Code auf, der getestet und über Ihre Pipeline bereitgestellt wird.
- Rollback-Mechanismus: Halten Sie stets eine einfache, zuverlässige und häufig getestete Methode bereit, um Änderungen rückgängig zu machen, falls etwas schief geht. Für die Sicherheit des Einsatzes ist es von entscheidender Bedeutung, schnell zum vorherigen, als funktionierend bekannten Zustand zurückkehren zu können. Dabei kann es sich um eine einfache Taste handeln, um zum vorherigen Zustand zurückzukehren, oder es kann vollständig automatisiert und durch Alarme ausgelöst werden.
- Versionskontrolle: Pflegen Sie den gesamten Anwendungscode, die Konfiguration und sogar die Infrastruktur als Code in einem versionskontrollierten Repository. Diese Vorgehensweise trägt dazu bei, dass Sie Änderungen einfach nachverfolgen und bei Bedarf rückgängig machen können.
- Kanarische Bereitstellungen und Bereitstellungen in Blau/Grün: Wenn Sie neue Versionen Ihrer Anwendung zunächst in einem Teil Ihrer Infrastruktur bereitstellen oder zwei Umgebungen (blau/grün) verwalten, können Sie das Verhalten einer Änderung in der Produktion überprüfen und bei Bedarf schnell ein Rollback durchführen.

Bei CI/CD geht es nicht nur um die Tools, sondern auch um die Kultur. Die Schaffung einer Kultur, die Wert auf Automatisierung, Testen und Lernen aus Fehlern legt, ist genauso wichtig wie die Implementierung der richtigen Tools und Prozesse. Rollbacks sollten, wenn sie sehr schnell und mit minimalen Auswirkungen durchgeführt werden, nicht als Misserfolg, sondern als Lernerfahrung betrachtet werden.

Durchführung von ORRs

Ein Operational Readiness Review (ORR) hilft dabei, betriebliche und verfahrenstechnische Lücken zu identifizieren. Bei Amazon haben wir ORRs entwickelt, um die Erkenntnisse aus Jahrzehnten des Betriebs hochwertiger Dienste in kuratierten Fragen mit Best-Practice-Anleitungen zusammenzufassen. Ein ORR erfasst frühere Erkenntnisse und verlangt von neuen Teams, sicherzustellen, dass sie diese Erkenntnisse in ihren Anwendungen berücksichtigt haben. ORRs können eine Liste von Ausfallarten oder Fehlerursachen bereitstellen, die in die im Abschnitt Resilienzmodellierung unten beschriebene Aktivität zur Resilienzmodellierung einfließen können. Weitere Informationen finden Sie unter Operational Readiness Reviews (ORRs) auf der AWS Well-Architected Framework-Website.

Die Grenzen der Fehlerisolierung verstehen AWS

AWS bietet mehrere Grenzen zur Fehlerisolierung, damit Sie Ihre Ausfallsicherheitsziele erreichen können. Sie können diese Grenzen nutzen, um den vorhersehbaren Umfang der Schadensbegrenzung zu nutzen, den sie bieten. Sie sollten mit der Gestaltung von AWS Diensten anhand dieser Grenzen vertraut sein, sodass Sie bewusst entscheiden können, welche Abhängigkeiten Sie für Ihre Anwendung auswählen. Informationen zur Verwendung von Grenzen in Ihrer Anwendung finden Sie auf der AWS Website unter AWS Fault Isolation Boundaries.

Antworten auswählen

Ein System kann auf vielfältige Weise auf einen Alarm reagieren. Einige Alarme erfordern möglicherweise eine Reaktion des Betriebsteams, während andere Selbstheilungsmechanismen innerhalb der Anwendung auslösen können. Möglicherweise entscheiden Sie sich dafür, Antworten, die automatisiert werden könnten, als manuelle Operationen beizubehalten, um die Kosten der Automatisierung zu kontrollieren oder technische Einschränkungen zu bewältigen. Die Art der Reaktion auf einen Alarm wird wahrscheinlich in Abhängigkeit von den Kosten für die Implementierung der Reaktion, der voraussichtlichen Häufigkeit des Alarms, der Genauigkeit des

Alarms und dem potenziellen Geschäftsverlust ausgewählt, wenn auf den Alarm überhaupt nicht reagiert wird.

Wenn beispielsweise ein Serverprozess abstürzt, kann der Prozess vom Betriebssystem neu gestartet werden, oder es kann ein neuer Server bereitgestellt und der alte beendet werden, oder ein Bediener kann angewiesen werden, eine Remoteverbindung mit dem Server herzustellen und ihn neu zu starten. Diese Reaktionen haben dasselbe Ergebnis, nämlich den Neustart des Anwendungsserverprozesses, haben jedoch unterschiedliche Implementierungs- und Wartungskosten zur Folge.

> **Note**
>
> Sie können mehrere Antworten auswählen, um einen umfassenden Resilienzansatz zu verfolgen. Im vorherigen Szenario könnte sich das Anwendungsteam beispielsweise dafür entscheiden, alle drei Antworten mit einer zeitlichen Verzögerung zwischen den einzelnen Antworten zu implementieren. Wenn sich die Prozessanzeige für ausgefallenen Server nach 30 Sekunden immer noch in einem Alarmzustand befindet, kann das Team davon ausgehen, dass das Betriebssystem den Anwendungsserver nicht neu starten konnte. Daher können sie eine Auto Scaling-Gruppe erstellen, um einen neuen virtuellen Server zu erstellen und den Anwendungsserverprozess wiederherzustellen. Wenn sich der Indikator nach 300 Sekunden immer noch im Alarmzustand befindet, wird möglicherweise eine Warnung an das Betriebspersonal gesendet, um eine Verbindung zum ursprünglichen Server herzustellen und zu versuchen, den Prozess wiederherzustellen.

Die Reaktion, die das Anwendungsteam und das Unternehmen wählen, sollte dem Wunsch des Unternehmens entsprechen, die betrieblichen Gemeinkosten durch Vorabinvestitionen in Entwicklungszeit auszugleichen. Sie sollten eine Antwort wählen — ein Architekturmuster wie statische Stabilität, ein Softwaremuster wie ein Schutzschalter oder ein Betriebsverfahren —, indem Sie die Einschränkungen und die erwartete Wartung der einzelnen Reaktionsoptionen sorgfältig abwägen. Möglicherweise gibt es einige Standardlösungen, die Anwendungsteams als Leitfaden dienen, sodass Sie die Bibliotheken und Muster, die von Ihrer zentralen Architekturfunktion verwaltet werden, als Grundlage für diese Überlegungen verwenden können.

Modellierung der Resilienz

Die Resilienzmodellierung dokumentiert, wie eine Anwendung auf verschiedene erwartete Störungen reagieren wird. Durch die Antizipation von Störungen kann Ihr Team Beobachtbarkeit, automatisierte

Kontrollen und Wiederherstellungsprozesse implementieren, um Beeinträchtigungen trotz Störungen zu minimieren oder zu verhindern. AWS hat Leitlinien für die Entwicklung eines Resilienzmodells unter Verwendung des Frameworks für Resilienzanalysen erstellt. Dieses Framework kann Ihnen helfen, Störungen und deren Auswirkungen auf Ihre Anwendung zu antizipieren. Durch die Antizipation von Störungen können Sie die Maßnahmen identifizieren, die für den Aufbau einer belastbaren, zuverlässigen Anwendung erforderlich sind. Wir empfehlen Ihnen, das Framework für die Resilienzanalyse zu verwenden, um Ihr Resilienzmodell bei jeder Iteration des Lebenszyklus Ihrer Anwendung zu aktualisieren. Die Verwendung dieses Frameworks bei jeder Iteration trägt dazu bei, Vorfälle zu reduzieren, indem Störungen während der Entwurfsphase antizipiert und die Anwendung vor und nach der Produktionsbereitstellung getestet wird. Durch die Entwicklung eines Resilienzmodells mithilfe dieses Frameworks können Sie sicherstellen, dass Sie Ihre Resilienzziele erreichen.

Sicher scheitern

Wenn Sie Störungen nicht vermeiden können, scheitern Sie sicher. Erwägen Sie, Ihre Anwendung mit einem standardmäßigen ausfallsicheren Betriebsmodus zu erstellen, in dem kein nennenswerter Geschäftsverlust entstehen kann. Ein Beispiel für einen ausfallsicheren Zustand einer Datenbank wäre die Standardeinstellung für schreibgeschützte Operationen, bei denen Benutzer keine Daten erstellen oder mutieren dürfen. Abhängig von der Vertraulichkeit der Daten möchten Sie vielleicht sogar, dass die Anwendung standardmäßig heruntergefahren wird und nicht einmal schreibgeschützte Abfragen durchführt. Überlegen Sie, in welchem Zustand Ihre Anwendung ausfallsicher sein sollte, und verwenden Sie unter extremen Bedingungen standardmäßig diesen Betriebsmodus.

Stufe 3: Auswerten und Testen

Während der Evaluierungs- und Testphase des Lebenszyklus wurden die Anwendung oder Änderungen an einer bestehenden Anwendung zwar entworfen, aber noch nicht für die Produktion freigegeben. In dieser Phase implementieren Sie Aktivitäten, um die Praktiken zu testen, die in früheren Phasen angewendet wurden, und um die Ergebnisse auszuwerten. Die Anwendung befindet sich möglicherweise noch in der aktiven Entwicklung, oder die Primärentwicklung ist abgeschlossen und die Anwendung wird möglicherweise getestet, bevor sie für die Produktion freigegeben wird. In dieser Phase konzentrieren Sie sich auf die Entwicklung und Durchführung von Tests, die die Erwartungen bestätigen oder widerlegen, dass die Anwendung die definierten Stabilitätsziele erfüllt. Darüber hinaus entwickeln und testen Sie die Betriebsabläufe des Systems. Die Bereitstellungsverfahren, die Sie in der Phase 2: Entwurf und Implementierung entwickelt haben, werden in die Praxis umgesetzt und die Ergebnisse werden bewertet. Diese Test- und Evaluierungsaktivitäten beginnen zwar in diesem Teil des Lebenszyklus, enden aber nicht hier. Die Tests und Evaluierungen werden fortgesetzt, während Sie in die Phase 4: Betrieb übergehen.

Die Evaluierungs- und Testphase ist in zwei Phasen unterteilt: Aktivitäten vor der Bereitstellung und Aktivitäten nach der Bereitstellung. Aktivitäten vor der Bereitstellung bestehen aus Aufgaben, die abgeschlossen werden müssen, bevor Sie die Anwendung in einer beliebigen Umgebung bereitstellen, einschließlich der Bereitstellung neuer Versionen der Software sowie der ersten Bereitstellung in einer Testumgebung. Aktivitäten nach der Bereitstellung finden statt, nachdem die Software in einer Test- oder Produktionsumgebung bereitgestellt wurde. In den folgenden Abschnitten werden diese Phasen ausführlicher behandelt.

Aktivitäten vor der Bereitstellung

Gestaltung der Umgebung

Die Umgebung, in der Sie Ihre Anwendung testen und bewerten, wirkt sich darauf aus, wie gründlich Sie sie testen können und wie sicher Sie sein können, dass diese Ergebnisse genau wiedergeben, was in der Produktion passieren wird. Möglicherweise können Sie mithilfe von Diensten wie Amazon DynamoDB einige Integrationstests lokal auf Entwicklercomputern durchführen (siehe DynamoDB lokal einrichten in der DynamoDB-Dokumentation). Irgendwann müssen Sie jedoch in einer Umgebung testen, die Ihre Produktionsumgebung repliziert, um ein Höchstmaß an Vertrauen in Ihre Ergebnisse zu erreichen. Da diese Umgebung mit Kosten verbunden ist, empfehlen wir Ihnen, für Ihre Umgebungen einen schrittweisen Ansatz oder eine Pipeline zu wählen, bei der produktionsmäßige Umgebungen erst später in der Pipeline auftauchen.

Integrationstests

Integrationstests sind der Prozess, bei dem getestet wird, ob eine genau definierte Komponente einer Anwendung ihre Funktionen korrekt ausführt, wenn sie mit externen Abhängigkeiten arbeitet. Bei diesen externen Abhängigkeiten kann es sich um andere individuell entwickelte Komponenten, AWS Dienste, die Sie für Ihre Anwendung verwenden, Abhängigkeiten von Drittanbietern und lokale Abhängigkeiten handeln. Dieser Leitfaden konzentriert sich auf Integrationstests, mit denen die Belastbarkeit Ihrer Anwendung nachgewiesen wird. Es wird davon ausgegangen, dass es bereits Einheiten- und Integrationstests gibt, die die Funktionsgenauigkeit Ihrer Software belegen.

Wir empfehlen Ihnen, Integrationstests zu entwerfen, die speziell die von Ihnen implementierten Resilienzmuster testen, z. B. Schutzschaltermuster oder Lastabwurf (siehe Phase 2: Design und Implementierung). Resilienzorientierte Integrationstests beinhalten häufig das Aufbringen einer bestimmten Last auf die Anwendung oder das gezielte Herbeiführen von Störungen in der Umgebung mithilfe von Funktionen wie ().AWS Fault Injection ServiceAWS FIS Idealerweise sollten Sie alle Integrationstests als Teil Ihrer CI/CD-Pipeline ausführen und sicherstellen, dass Sie die Tests jedes Mal ausführen, wenn der Code festgeschrieben wird. Auf diese Weise können Sie Änderungen am Code oder an Konfigurationen, die zu Verstößen gegen Ihre Resilienzziele führen, schnell erkennen und darauf reagieren. Umfangreiche verteilte Anwendungen sind komplex, und selbst geringfügige Änderungen können die Widerstandsfähigkeit scheinbar unzusammenhängender Teile Ihrer Anwendung erheblich beeinträchtigen. Versuchen Sie, Ihre Tests bei jedem Commit auszuführen. AWS bietet hervorragende Tools für den Betrieb Ihrer CI/CD-Pipeline und anderer DevOps Tools. Weitere Informationen finden Sie unter Einführung in DevOps on AWS auf der AWS Website.

Automatisierte Bereitstellungspipelines

Die Bereitstellung in Ihren Vorproduktionsumgebungen und das Testen in diesen Umgebungen ist eine sich wiederholende und komplexe Aufgabe, die am besten der Automatisierung überlassen wird. Durch die Automatisierung dieses Prozesses werden Personalressourcen entlastet und die Fehleranfälligkeit verringert. Der Mechanismus zur Automatisierung dieses Prozesses wird oft als Pipeline bezeichnet. Wenn Sie Ihre Pipeline erstellen, empfehlen wir Ihnen, eine Reihe von Testumgebungen einzurichten, die Ihrer Produktionskonfiguration immer näher kommen. Sie verwenden diese Reihe von Umgebungen, um Ihre Anwendung wiederholt zu testen. Die erste Umgebung bietet einen eingeschränkteren Funktionsumfang als die Produktionsumgebung, ist jedoch mit deutlich geringeren Kosten verbunden. In nachfolgenden Umgebungen sollten Dienste hinzugefügt und skaliert werden, damit sie der Produktionsumgebung besser entsprechen.

Beginnen Sie mit dem Testen in der ersten Umgebung. Nachdem Ihre Bereitstellungen alle Tests in der ersten Testumgebung bestanden haben, lassen Sie die Anwendung für einen bestimmten Zeitraum unter einer gewissen Last laufen, um festzustellen, ob im Laufe der Zeit Probleme auftreten. Vergewissern Sie sich, dass Sie die Beobachtbarkeit korrekt konfiguriert haben (siehe Alarmgenauigkeit weiter unten in diesem Handbuch), damit Sie alle auftretenden Probleme erkennen können. Wenn dieser Beobachtungszeitraum erfolgreich abgeschlossen ist, stellen Sie Ihre Anwendung in Ihrer nächsten Testumgebung bereit und wiederholen Sie den Vorgang, indem Sie zusätzliche Tests oder Lasten hinzufügen, je nachdem, wie von der Umgebung unterstützt. Nachdem Sie Ihre Anwendung auf diese Weise ausreichend getestet haben, können Sie die zuvor eingerichteten Bereitstellungsmethoden verwenden, um die Anwendung in der Produktion bereitzustellen (siehe Definieren von CI/CD-Strategien weiter oben in diesem Handbuch). Der Artikel Automatisieren sicherer, automatischer Bereitstellungen in der Amazon Builders' Library ist eine hervorragende Ressource, die beschreibt, wie Amazon die Codebereitstellung automatisiert. Die Anzahl der Umgebungen, die Ihrer Produktionsbereitstellung vorausgehen, hängt von der Komplexität Ihrer Anwendung und den Arten der Abhängigkeiten ab.

Lasttest

Oberflächlich betrachtet ähneln Belastungstests Integrationstests. Sie testen eine einzelne Funktion Ihrer Anwendung und ihrer externen Abhängigkeiten, um sicherzustellen, dass sie erwartungsgemäß funktioniert. Der Lasttest geht dann über Integrationstests hinaus und konzentriert sich darauf, wie die Anwendung unter genau definierten Lasten funktioniert. Lasttests erfordern die Überprüfung der korrekten Funktionalität und müssen daher nach einem erfolgreichen Integrationstest durchgeführt werden. Es ist wichtig zu verstehen, wie gut die Anwendung unter erwarteten Belastungen reagiert und wie sie sich verhält, wenn die Last die Erwartungen übertrifft. Auf diese Weise können Sie überprüfen, ob Sie die erforderlichen Mechanismen implementiert haben, um sicherzustellen, dass Ihre Anwendung auch unter extremer Belastung widerstandsfähig bleibt. Eine umfassende Anleitung zum Testen von Lasten finden Sie unter Distributed Load Testing on AWS in der AWS Lösungsbibliothek. AWS

Aktivitäten nach der Bereitstellung

Resilienz ist ein fortlaufender Prozess, und die Bewertung der Belastbarkeit Ihrer Anwendung muss auch nach der Bereitstellung der Anwendung fortgesetzt werden. Die Ergebnisse Ihrer Aktivitäten nach der Bereitstellung, wie z. B. laufende Resilienzbewertungen, erfordern möglicherweise, dass Sie einige der Resilienzaktivitäten, die Sie zu Beginn des Resilienz-Lebenszyklus durchgeführt haben, neu bewerten und aktualisieren.

Durchführung von Resilienzanalysen

Die Bewertung der Resilienz hört nicht auf, nachdem Sie Ihre Anwendung in der Produktion bereitgestellt haben. Selbst wenn Sie über klar definierte und automatisierte Bereitstellungspipelines verfügen, können Änderungen manchmal direkt in einer Produktionsumgebung vorgenommen werden. Darüber hinaus kann es Faktoren geben, die Sie bei der Überprüfung der Belastbarkeit vor der Bereitstellung noch nicht berücksichtigt haben. AWS Resilience Hubbietet einen zentralen Ort, an dem Sie beurteilen können, ob Ihre bereitgestellte Architektur Ihren definierten RPO- und RTO-Anforderungen entspricht. Sie können diesen Service verwenden, um auf Abruf die Ausfallsicherheit Ihrer Anwendung zu bewerten, Bewertungen zu automatisieren und sie sogar in Ihre CI/CD-Tools zu integrieren, wie im AWS Blogbeitrag Kontinuierliche Bewertung der Anwendungsausfallsicherheit mit und beschrieben. AWS Resilience Hub AWS CodePipeline Die Automatisierung dieser Bewertungen ist eine bewährte Methode, da sie sicherstellt, dass Sie Ihre Ausfallsicherheit in der Produktion kontinuierlich bewerten.

DR-Tests

In Phase 2: Design und Implementierung haben Sie Disaster Recovery-Strategien (DR) als Teil Ihres Systems entwickelt. In Phase 4 sollten Sie Ihre DR-Verfahren testen, um sicherzustellen, dass Ihr Team auf einen Vorfall umfassend vorbereitet ist und Ihre Verfahren erwartungsgemäß funktionieren. Sie sollten alle Ihre DR-Verfahren, einschließlich Failover und Failback, regelmäßig testen und die Ergebnisse jeder Übung überprüfen, um festzustellen, ob und wie die Verfahren Ihres Systems aktualisiert werden sollten, um das bestmögliche Ergebnis zu erzielen. Wenn Sie Ihren DR-Test zu Beginn entwickeln, sollten Sie den Test rechtzeitig planen und sicherstellen, dass das gesamte Team weiß, was zu erwarten ist, wie die Ergebnisse gemessen werden und welcher Feedback-Mechanismus verwendet wird, um die Verfahren auf der Grundlage der Ergebnisse zu aktualisieren. Nachdem Sie sich mit der Durchführung von geplanten DR-Tests vertraut gemacht haben, sollten Sie die Durchführung unangekündigter DR-Tests in Betracht ziehen. Echte Katastrophen ereignen sich nicht nach einem bestimmten Zeitplan. Sie müssen also darauf vorbereitet sein, Ihren Plan jederzeit in die Tat umzusetzen. Unangekündigt bedeutet jedoch nicht ungeplant. Die wichtigsten Beteiligten müssen die Veranstaltung weiterhin planen, um sicherzustellen, dass eine angemessene Überwachung erfolgt und dass Kunden und kritische Anwendungen nicht beeinträchtigt werden.

Erkennung von Abweichungen

Unvorhergesehene Änderungen an der Konfiguration von Produktionsanwendungen können auftreten, selbst wenn Automatisierung und klar definierte Verfahren vorhanden sind. Um Änderungen an der Konfiguration Ihrer Anwendung erkennen zu können, sollten Sie über

Mechanismen zur Erkennung von Abweichungen verfügen. Dabei handelt es sich um Abweichungen von einer Basiskonfiguration. Informationen dazu, wie Sie Abweichungen in Ihren AWS CloudFormation Stacks erkennen können, finden Sie in der Dokumentation unter Erkennen nicht verwalteter Konfigurationsänderungen an Stacks und Ressourcen. AWS CloudFormation Informationen zur Erkennung von Abweichungen in der AWS Umgebung Ihrer Anwendung finden Sie AWS Control Tower in der Dokumentation unter Abweichungen erkennen und beheben. AWS Control Tower

Synthetisches Testen

Synthetisches Testen ist der Prozess der Erstellung konfigurierbarer Software, die nach einem Zeitplan in der Produktion ausgeführt wird, um die APIs Ihrer Anwendung so zu testen, dass die Endbenutzererfahrung simuliert wird. Diese Tests werden manchmal als Kanarienvögel bezeichnet, was auf die ursprüngliche Verwendung des Begriffs im Kohlebergbau zurückzuführen ist. Synthetische Tests können häufig frühzeitig warnen, wenn es bei einer Anwendung zu Störungen kommt, auch wenn die Beeinträchtigung nur teilweise oder zeitweise erfolgt, wie dies häufig bei grauen Fehlern der Fall ist.

Chaos-Technik

Chaos Engineering ist ein systematischer Prozess, bei dem eine Anwendung bewusst und auf risikomindernde Weise störenden Ereignissen ausgesetzt, ihre Reaktion genau überwacht und notwendige Verbesserungen umgesetzt werden. Sein Zweck besteht darin, Annahmen über die Fähigkeit der Anwendung, mit solchen Störungen umzugehen, zu validieren oder in Frage zu stellen. Anstatt diese Ereignisse dem Zufall zu überlassen, ermöglicht Chaos Engineering den Ingenieuren die Durchführung von Experimenten in einer kontrollierten Umgebung, typischerweise in Zeiten mit geringem Verkehrsaufkommen, und mit leicht verfügbarer technischer Unterstützung für eine wirksame Eindämmung.

Chaos Engineering beginnt mit dem Verständnis der normalen Betriebsbedingungen, des sogenannten stationären Zustands, der betrachteten Anwendung. Auf dieser Grundlage formulieren Sie eine Hypothese, in der das erfolgreiche Verhalten der Anwendung bei Störungen detailliert beschrieben wird. Sie führen das Experiment durch, bei dem bewusst Störungen wie Netzwerklatenz, Serverausfälle, Festplattenfehler und Beeinträchtigung externer Abhängigkeiten eingeführt werden. Anschließend analysieren Sie die Ergebnisse des Experiments und verbessern die Widerstandsfähigkeit der Anwendung auf der Grundlage Ihrer Erkenntnisse. Das Experiment dient als wertvolles Instrument zur Verbesserung verschiedener Aspekte der Anwendung, einschließlich ihrer Leistung, und deckt latente Probleme auf, die andernfalls möglicherweise verborgen geblieben

wären. Darüber hinaus hilft Chaos Engineering dabei, Mängel bei Tools zur Beobachtbarkeit und Alarmierung aufzudecken und diese zu verfeinern. Es trägt auch dazu bei, die Wiederherstellungszeit zu verkürzen und die operativen Fähigkeiten zu verbessern. Chaos Engineering beschleunigt die Einführung von Best Practices und fördert eine Einstellung zur kontinuierlichen Verbesserung. Letztlich ermöglicht es Teams, ihre operativen Fähigkeiten durch regelmäßiges Üben und Wiederholen aufzubauen und zu verbessern.

AWS empfiehlt, dass Sie Ihre Chaos-Engineering-Bemühungen in einer Umgebung beginnen, die nicht zur Produktion gehört. Sie können AWS Fault Injection Service (AWS FIS) verwenden, um Experimente zur Chaos-Technik durchzuführen, bei denen es sich sowohl um allgemeine Fehler als auch um Fehler handelt, die nur bei auftreten. AWS Dieser vollständig verwaltete Service umfasst Alarme bei Stoppen und umfassende Genehmigungskontrollen, sodass Sie Chaos Engineering problemlos und sicher einsetzen können.

Stufe 4: Bedienen

Nachdem Sie Phase 3: Evaluieren und Testen abgeschlossen haben, sind Sie bereit, die Anwendung in der Produktion bereitzustellen. In der Operate-Phase stellen Sie Ihre Anwendung in der Produktion bereit und verwalten das Kundenerlebnis. Der Entwurf und die Implementierung Ihrer Anwendung bestimmen viele der Ergebnisse der Ausfallsicherheit. In dieser Phase stehen jedoch die betrieblichen Verfahren im Mittelpunkt, mit denen Ihr System die Ausfallsicherheit aufrechterhält und verbessert. Der Aufbau einer Kultur betrieblicher Exzellenz trägt dazu bei, Standards und Konsistenz in diesen Verfahren zu schaffen.

Beobachtbarkeit

Der wichtigste Teil des Verständnisses des Kundenerlebnisses besteht in der Überwachung und Alarmierung. Sie müssen Ihre Anwendung instrumentieren, um ihren Status zu verstehen, und Sie benötigen unterschiedliche Perspektiven, was bedeutet, dass Sie sowohl von der Server- als auch von der Clientseite aus messen müssen, was in der Regel bei Kanarienvögeln der Fall ist. Ihre Messwerte sollten Daten über die Interaktionen Ihrer Anwendung mit ihren Abhängigkeiten und Dimensionen enthalten, die Ihren Grenzen zur Fehlerisolierung entsprechen. Sie sollten auch Protokolle erstellen, die zusätzliche Details zu jeder von Ihrer Anwendung ausgeführten Arbeitseinheit enthalten. Sie könnten erwägen, Metriken und Protokolle zu kombinieren, indem Sie eine Lösung wie das CloudWatcheingebettete Metrikformat von Amazon verwenden. Sie werden wahrscheinlich feststellen, dass Sie immer mehr Beobachtbarkeit wünschen. Berücksichtigen Sie daher die Kompromisse zwischen Kosten, Aufwand und Komplexität, die für die Implementierung der gewünschten Instrumentierung erforderlich sind.

Die folgenden Links bieten bewährte Methoden für die Instrumentierung Ihrer Anwendung und die Erstellung von Alarmen:

- Überwachung von Produktionsdienstleistungen bei Amazon (Präsentation AWS re:Invent 2020)
- Amazon Builders' Library: Operational Excellence bei Amazon (Präsentation AWS re:Invent 2021)
- Bewährte Methoden zur Beobachtbarkeit bei Amazon (Präsentation AWS re:Invent 2022)
- Instrumentierung verteilter Systeme für betriebliche Transparenz (Artikel in der Amazon Builders' Library)
- Erstellung von Dashboards für betriebliche Transparenz (Artikel in der Amazon Builders' Library)

Verwaltung von Veranstaltungen

Sie sollten über einen Event-Management-Prozess verfügen, um mit Beeinträchtigungen umzugehen, wenn Ihre Alarmanlagen (oder schlimmer noch, Ihre Kunden) Ihnen mitteilen, dass etwas schief läuft. Dieser Prozess sollte die Beauftragung eines Bereitschaftsdienstmitarbeiters, die Eskalation von Problemen und die Einrichtung von Runbooks für konsistente Lösungsansätze beinhalten, mit denen menschliche Fehler vermieden werden können. Beeinträchtigungen treten jedoch in der Regel nicht isoliert auf. Eine einzelne Anwendung kann sich auf mehrere andere Anwendungen auswirken, die von ihr abhängig sind. Sie können Probleme schnell lösen, indem Sie alle betroffenen Anwendungen verstehen und Mitarbeiter aus mehreren Teams in einer einzigen Telefonkonferenz zusammenbringen. Je nach Größe und Struktur Ihres Unternehmens kann dieser Prozess jedoch ein zentralisiertes Betriebsteam erfordern.

Zusätzlich zur Einrichtung eines Event-Management-Prozesses sollten Sie Ihre Kennzahlen regelmäßig anhand von Dashboards überprüfen. Regelmäßige Überprüfungen helfen Ihnen dabei, das Kundenerlebnis und die längerfristigen Trends bei der Leistung Ihrer Anwendung besser zu verstehen. Auf diese Weise können Sie Probleme und Engpässe erkennen, bevor sie erhebliche Auswirkungen auf die Produktion haben. Eine konsistente, standardisierte Überprüfung der Kennzahlen bietet erhebliche Vorteile, erfordert jedoch die Zustimmung von oben nach unten und einen hohen Zeitaufwand.

Die folgenden Links bieten bewährte Methoden zur Erstellung von Dashboards und zur Überprüfung betrieblicher Kennzahlen:

- Erstellung von Dashboards für betriebliche Transparenz (Artikel in der Amazon Builders' Library)
- Der Ansatz von Amazon, erfolgreich zu scheitern (Präsentation AWS re:Invent 2019)

Kontinuierliche Belastbarkeit

In Phase 2: Design und Implementierung und Phase 3: Evaluieren und Testen haben Sie Überprüfungs- und Testaktivitäten eingeleitet, bevor Sie Ihre Anwendung in der Produktion einsetzen. Während der Betriebsphase sollten Sie diese Aktivitäten in der Produktion weiter wiederholen. Sie sollten die Resilienz Ihrer Anwendung regelmäßig anhand von AWS Well-Architected Framework-Reviews, Operational Readiness Readiness Readiness Reviews (ORRs) und dem Resilienzanalyse-Framework überprüfen. Auf diese Weise können Sie sicherstellen, dass Ihre Anwendung nicht von den etablierten Grundlagen und Standards abweicht, und Sie bleiben über neue oder aktualisierte

Leitlinien auf dem Laufenden. Diese kontinuierlichen Resilienzaktivitäten helfen Ihnen dabei, bisher unerwartete Störungen zu entdecken und neue Abhilfemaßnahmen zu finden.

Vielleicht möchten Sie auch erwägen, Spieltage und Experimente zur Chaos-Technik in der Produktion durchzuführen, nachdem Sie sie erfolgreich in Vorproduktionsumgebungen durchgeführt haben. An Spieltagen werden bekannte Ereignisse simuliert, zu deren Abmilderung Sie Resilienzmechanismen eingerichtet haben. An einem Spieltag könnte beispielsweise eine regionale Beeinträchtigung des Dienstes simuliert und ein AWS regionsübergreifendes Failover implementiert werden. Die Implementierung dieser Aktivitäten kann zwar mit erheblichem Aufwand verbunden sein, aber beide Methoden helfen Ihnen dabei, die Gewissheit zu gewinnen, dass Ihr System den Ausfallarten standhält, für die Sie es konzipiert haben.

Durch den Betrieb Ihrer Anwendungen, die Beobachtung betrieblicher Ereignisse, die Überprüfung von Kennzahlen und das Testen Ihrer Anwendung bieten sich Ihnen zahlreiche Gelegenheiten, darauf zu reagieren und zu lernen.

Stufe 5: Reagieren und lernen

Die Art und Weise, wie Ihre Anwendung auf störende Ereignisse reagiert, beeinflusst ihre Zuverlässigkeit. Es ist auch wichtig, aus Erfahrungen zu lernen und zu erfahren, wie Ihre Anwendung in der Vergangenheit auf Störungen reagiert hat, um ihre Zuverlässigkeit zu verbessern.

Die Phase „Reagieren und Lernen“ konzentriert sich auf Methoden, die Sie implementieren können, um besser auf störende Ereignisse in Ihren Anwendungen reagieren zu können. Sie umfasst auch Methoden, die Ihnen helfen, aus den Erfahrungen Ihrer Betriebsteams und Techniker den größtmöglichen Lernerfolg zu ziehen.

Erstellung von Berichten zur Analyse von Vorfällen

Wenn ein Vorfall eintritt, besteht die erste Maßnahme darin, weitere Schäden für Kunden und das Unternehmen so schnell wie möglich zu verhindern. Nach der Wiederherstellung der Anwendung besteht der nächste Schritt darin, zu verstehen, was passiert ist, und Maßnahmen zu identifizieren, um ein erneutes Auftreten zu verhindern. Diese Analyse nach dem Vorfall wird in der Regel in Form eines Berichts erfasst, in dem die Ereignisse, die zu einer Beeinträchtigung der Anwendung geführt haben, sowie die Auswirkungen der Unterbrechung auf die Anwendung, die Kunden und das Unternehmen dokumentiert werden. Solche Berichte werden zu wertvollen Lernartefakten und sollten unternehmensweit verbreitet werden.

> **Note**
>
> Es ist wichtig, Vorfälle ohne Schuldzuweisung zu analysieren. Gehen Sie davon aus, dass alle Betreiber aufgrund der ihnen zur Verfügung stehenden Informationen die beste und geeignetste Vorgehensweise gewählt haben. Verwenden Sie in einem Bericht nicht die Namen von Bedienern oder Technikern. Wenn menschliches Versagen als Grund für die Beeinträchtigung angeführt wird, müssen Teammitglieder möglicherweise vorsichtig sein, um sich selbst zu schützen, was zur Erfassung falscher oder unvollständiger Informationen führen kann.

Ein guter Bericht zur Vorfallanalyse, wie er im Amazon Correction of Error (COE) -Prozess dokumentiert ist, folgt einem standardisierten Format und versucht, die Bedingungen, die zu einer Beeinträchtigung der Anwendung geführt haben, so detailliert wie möglich zu erfassen. Der

Bericht beschreibt eine Reihe von Ereignissen mit Zeitstempel und erfasst quantitative Daten (häufig Kennzahlen und Screenshots aus Monitoring-Dashboards), die den messbaren Status der Anwendung im Laufe des Zeitrahmens beschreiben. In dem Bericht sollten die Denkprozesse der Bediener und Techniker, die Maßnahmen ergriffen haben, sowie die Informationen, die sie zu ihren Schlussfolgerungen geführt haben, erfasst werden. In dem Bericht sollte auch die Leistung der verschiedenen Indikatoren detailliert beschrieben werden, z. B. welche Alarme ausgelöst wurden, ob diese Alarme den Status der Anwendung genau widerspiegelten, wie viel Zeit zwischen den Ereignissen und den daraus resultierenden Alarmen vergeht und wie lange es dauert, bis der Vorfall behoben ist. In der Zeitleiste werden auch die Runbooks oder Automatisierungen erfasst, die initiiert wurden, und wie sie dazu beigetragen haben, dass die Anwendung wieder in einen brauchbaren Zustand zurückversetzt wurde. Diese Elemente des Zeitplans helfen Ihrem Team dabei, die Effektivität automatisierter und operativer Reaktionen zu verstehen, einschließlich der Frage, wie schnell das Problem behoben wurde und wie wirksam sie zur Minimierung der Störung beigetragen haben.

Dieses detaillierte Bild eines historischen Ereignisses ist ein wirksames Lehrmittel. Teams sollten diese Berichte in einem zentralen Repository speichern, das dem gesamten Unternehmen zur Verfügung steht, damit andere die Ereignisse überprüfen und daraus lernen können. Dies kann die Intuition Ihrer Teams dafür verbessern, was in der Produktion schief gehen kann.

Eine Sammlung detaillierter Vorfallberichte wird auch zu einer Quelle für Schulungsmaterial für Bediener. Teams können einen Zwischenfallbericht als Inspiration für einen Spieltag am Tisch oder live nutzen, an dem die Teams Informationen erhalten, die den Zeitplan wiedergeben, der im Bericht aufgezeichnet ist. Die Bediener können das Szenario anhand von Teilinformationen aus der Zeitleiste durchgehen und beschreiben, welche Maßnahmen sie ergreifen würden. Der Moderator für den Spieltag kann dann anhand der Aktionen des Operators erläutern, wie die Anwendung reagiert hat. Dadurch entwickeln sich die Fähigkeiten der Bediener zur Problembehebung, sodass sie Probleme leichter antizipieren und beheben können.

Ein zentrales Team, das für die Zuverlässigkeit der Anwendungen verantwortlich ist, sollte diese Berichte in einer zentralen Bibliothek verwalten, auf die das gesamte Unternehmen zugreifen kann. Dieses Team sollte auch für die Pflege der Berichtsvorlage verantwortlich sein und die Teams darin schulen, wie der Bericht zur Vorfallanalyse auszufüllen ist. Das Zuverlässigkeitsteam sollte die Berichte regelmäßig überprüfen, um Trends im gesamten Unternehmen zu erkennen, denen durch Softwarebibliotheken, Architekturmuster oder Änderungen der Teamprozesse begegnet werden kann.

Durchführung betrieblicher Überprüfungen

Wie in Phase 4: Operate erörtert, bieten Betriebsüberprüfungen die Gelegenheit, aktuelle Funktionen, Vorfälle und Betriebskennzahlen zu überprüfen. Die operative Überprüfung bietet auch die Gelegenheit, Erkenntnisse aus neuen Funktionen und Vorfällen mit der gesamten technischen Community in Ihrem Unternehmen zu teilen. Während der operativen Überprüfung überprüfen die Teams die Bereitstellung von Funktionen, für die ein Rollback vorgenommen wurde, die aufgetretenen Vorfälle und die Art und Weise, wie sie behandelt wurden. Dies gibt Technikern im gesamten Unternehmen die Möglichkeit, aus den Erfahrungen anderer zu lernen und Fragen zu stellen.

Stellen Sie Ihre Betriebsbeurteilungen der technischen Community in Ihrem Unternehmen zur Verfügung, damit diese mehr über die IT-Anwendungen, die das Unternehmen betreiben, und über die Arten von Problemen, auf die sie stoßen können, erfahren können. Sie nehmen dieses Wissen mit, wenn sie andere Anwendungen für das Unternehmen entwerfen, implementieren und bereitstellen.

Überprüfung der Alarmleistung

Alarme können, wie in der Betriebsphase beschrieben, dazu führen, dass Warnmeldungen im Dashboard angezeigt, Tickets erstellt, E-Mails gesendet oder Bediener angerufen werden. Für eine Anwendung werden zahlreiche Alarme konfiguriert, um verschiedene Aspekte ihres Betriebs zu überwachen. Im Laufe der Zeit sollten die Genauigkeit und Wirksamkeit dieser Alarme überprüft werden, um die Alarmgenauigkeit zu erhöhen, Fehlalarme zu reduzieren und doppelte Warnmeldungen zu konsolidieren.

Präzision der Alarme

Alarme sollten so spezifisch wie möglich sein, um den Zeitaufwand für die Interpretation oder Diagnose der spezifischen Störung, die den Alarm ausgelöst hat, zu reduzieren. Wenn als Reaktion auf eine Beeinträchtigung der Anwendung ein Alarm ausgelöst wird, müssen die Bediener, die den Alarm empfangen und darauf reagieren, zunächst die Informationen interpretieren, die der Alarm übermittelt. Bei den Informationen kann es sich um einen einfachen Fehlercode handeln, der einer Vorgehensweise wie einem Wiederherstellungsverfahren entspricht, oder sie können Zeilen aus Anwendungsprotokollen enthalten, die Sie überprüfen müssen, um zu verstehen, warum der Alarm ausgelöst wurde. Wenn Ihr Team lernt, eine Anwendung effektiver zu bedienen, sollte es diese Alarme verfeinern, um sie so klar und präzise wie möglich zu gestalten.

Sie können nicht alle möglichen Störungen einer Anwendung vorhersehen. Daher wird es immer allgemeine Alarme geben, die von einem Bediener analysiert und diagnostiziert werden müssen. Ihr Team sollte daran arbeiten, die Anzahl der allgemeinen Alarme zu reduzieren, um die Reaktionszeiten zu verbessern und die mittlere Reparaturzeit (MTTR) zu verkürzen. Idealerweise sollte ein one-to-one Zusammenhang zwischen einem Alarm und einer automatisierten oder von Menschen ausgeführten Reaktion bestehen.

Falsch positive Ergebnisse

Alarme, bei denen kein Eingreifen des Bedieners erforderlich ist, sondern Warnmeldungen in Form von E-Mails, Seiten oder Tickets generiert werden, werden von den Bedienern im Laufe der Zeit ignoriert. Überprüfen Sie in regelmäßigen Abständen oder im Rahmen einer Vorfallanalyse die Alarme, um festzustellen, welche Alarme häufig ignoriert werden oder keine Maßnahmen seitens der Bediener erfordern (Fehlalarme). Sie sollten daran arbeiten, entweder den Alarm zu entfernen oder den Alarm so zu verbessern, dass er eine umsetzbare Warnung an die Bediener ausgibt.

Falsch negative Ergebnisse

Während eines Vorfalls können Alarme, die so konfiguriert sind, dass sie während des Vorfalls warnen, fehlschlagen, möglicherweise aufgrund eines Ereignisses, das sich unerwartet auf die Anwendung auswirkt. Im Rahmen einer Vorfallanalyse sollten Sie die Alarme überprüfen, die hätten ausgelöst werden sollen, aber nicht ausgelöst wurden. Sie sollten daran arbeiten, diese Alarme so zu verbessern, dass sie die Bedingungen, die sich aus einem Ereignis ergeben könnten, besser widerspiegeln. Alternativ müssen Sie möglicherweise zusätzliche Alarme erstellen, die derselben Störung zugeordnet sind, aber durch ein anderes Symptom der Störung ausgelöst werden.

Doppelte Warnmeldungen

Eine Störung, die Ihre Anwendung beeinträchtigt, verursacht wahrscheinlich mehrere Symptome und kann zu mehreren Alarmen führen. In regelmäßigen Abständen oder im Rahmen einer Vorfallanalyse sollten Sie die ausgegebenen Alarme und Warnmeldungen überprüfen. Wenn Bediener doppelte Warnmeldungen erhalten haben, erstellen Sie aggregierte Alarme, um sie in einer einzigen Warnmeldung zusammenzufassen.

Durchführung von Überprüfungen von Kennzahlen

Ihr Team sollte betriebliche Kennzahlen zu Ihrer Anwendung erheben, z. B. die Anzahl der Vorfälle nach Schweregrad pro Monat, die Zeit bis zur Erkennung des Vorfalls, die Zeit bis zur Identifizierung

der Ursache, die Zeit bis zur Behebung und die Anzahl der erstellten Tickets, gesendeten Benachrichtigungen und aufgerufenen Seiten. Überprüfen Sie diese Kennzahlen mindestens einmal im Monat, um zu verstehen, mit welcher Belastung das Betriebspersonal konfrontiert ist, mit welchem signal-to-noise Verhältnis es zu tun hat (z. B. informative und umsetzbare Warnmeldungen) und ob das Team seine Fähigkeit verbessert, die von ihnen kontrollierten Anwendungen zu betreiben. Anhand dieser Übersicht können Sie sich ein Bild von Trends in den messbaren Bereichen des Betriebsteams machen. Bitten Sie das Team um Ideen zur Verbesserung dieser Kennzahlen.

Bereitstellung von Schulungen und Befähigungsmaßnahmen

Es ist schwierig, eine detaillierte Beschreibung einer Anwendung und ihrer Umgebung zu erfassen, die zu einem Vorfall oder unerwartetem Verhalten geführt haben. Darüber hinaus ist es nicht immer einfach, die Widerstandsfähigkeit Ihrer Anwendung zu modellieren, um solche Szenarien zu antizipieren. Ihr Unternehmen sollte in Schulungs- und Schulungsmaterial investieren, damit Ihre Betriebsteams und Entwickler an Aktivitäten wie Resilienzmodellierung, Störfallanalysen, Spieltagen und Experimenten zur Chaostechnik teilnehmen können. Dadurch werden die Genauigkeit der von Ihren Teams erstellten Berichte und das von ihnen erfasste Wissen verbessert. Die Teams werden auch besser in der Lage sein, Ausfälle zu antizipieren, ohne sich auf eine kleinere, erfahrenere Gruppe von Technikern verlassen zu müssen, die ihre Erkenntnisse im Rahmen von geplanten Überprüfungen einbringen müssen.

Aufbau einer Wissensdatenbank zu Vorfällen

Ein Vorfallbericht ist eine Standardausgabe einer Vorfallanalyse. Sie sollten denselben oder einen ähnlichen Bericht verwenden, um Szenarien zu dokumentieren, in denen Sie ein ungewöhnliches Anwendungsverhalten festgestellt haben, auch wenn die Anwendung nicht beeinträchtigt wurde. Verwenden Sie dieselbe standardisierte Berichtsstruktur, um die Ergebnisse von Chaosexperimenten und Spieltagen zu erfassen. Der Bericht stellt eine Momentaufnahme der Anwendung und ihrer Umgebung dar, die zu einem Vorfall oder einem anderen unerwarteten Verhalten geführt haben. Sie sollten diese standardisierten Berichte in einem zentralen Repository speichern, auf das alle Techniker im Unternehmen zugreifen können.

Betriebsteams und Entwickler können dann diese Wissensdatenbank durchsuchen, um zu verstehen, was in der Vergangenheit zu Störungen bei Anwendungen geführt hat, welche Arten von Szenarien zu Störungen geführt haben könnten und was die Beeinträchtigung von Anwendungen verhindert hat. Diese Wissensdatenbank beschleunigt die Verbesserung der Fähigkeiten Ihrer Betriebsteams und Ihrer Entwickler und ermöglicht es ihnen, ihr Wissen und ihre Erfahrungen auszutauschen.

Darüber hinaus können Sie die Berichte als Schulungsmaterial oder als Szenarien für Spieltage oder Chaosexperimente verwenden, um die Intuition und Fähigkeit des operativen Teams zu verbessern, Störungen zu beheben.

> **Note**
>
> Ein standardisiertes Berichtsformat vermittelt den Lesern zudem ein Gefühl der Vertrautheit und hilft ihnen, die Informationen, nach denen sie suchen, schneller zu finden.

Umfassende Umsetzung von Resilienz

Wie bereits erwähnt, wird eine fortgeschrittene Organisation mehrere Maßnahmen ergreifen, um auf einen Alarm zu reagieren. Es gibt keine Garantie dafür, dass eine Reaktion wirksam ist. Durch die Kombination mehrerer Antworten ist eine Anwendung also besser darauf vorbereitet, fehlerhaft auszufallen. Wir empfehlen, für jeden Indikator mindestens zwei Antworten zu implementieren, um sicherzustellen, dass eine einzelne Reaktion nicht zu einer einzigen Fehlerquelle wird, die zu einem DR-Szenario führen könnte. Diese Ebenen sollten in serieller Reihenfolge erstellt werden, sodass eine aufeinanderfolgende Reaktion nur dann ausgeführt wird, wenn die vorherige Reaktion unwirksam war. Sie sollten nicht mehrere mehrschichtige Antworten auf einen einzelnen Alarm ausführen. Verwenden Sie stattdessen einen Alarm, der anzeigt, ob eine Reaktion nicht erfolgreich war, und wenn ja, die nächste mehrschichtige Reaktion einleitet.

Fazit und Ressourcen

Dieser Leitfaden stellt einen Lebenszyklus vor, der Sie dabei unterstützt, die Widerstandsfähigkeit Ihrer Anwendungen kontinuierlich zu verbessern, indem Sie bewährte Verfahren in fünf Phasen implementieren: Ziele festlegen, entwerfen und implementieren, evaluieren und testen, betreiben und reagieren und lernen.

Weitere Informationen zu den in diesem Leitfaden behandelten Services und Konzepten finden Sie in den folgenden Ressourcen.

AWS Dienste:

- AWS Backup
- AWS Elastic Disaster Recovery
- AWS Fault Injection Service (AWS FIS)
- AWS Resilience Hub
- Amazon Route 53-Controller für die Anwendungswiederherstellung
- AWS X-Ray

Blogbeiträge und Artikel:

- Verfügbarkeit und mehr: Verständnis und Verbesserung der Widerstandsfähigkeit verteilter Systeme auf AWS
- AWS Grenzen der Fehlerisolierung
- AWS Grundlagen für mehrere Regionen
- Chaos Engineering in der Cloud
- Kontinuierliche Bewertung der Widerstandsfähigkeit von Anwendungen mit AWS Resilience Hub und AWS CodePipeline
- Disaster Recovery von lokalen Anwendungen zu AWS
- Säule der Zuverlässigkeit — AWS Well-Architected Framework
- Rahmen für die Resilienzanalyse

Mitwirkende

Zu den Mitwirkenden an diesem Leitfaden gehören:

- Bruno Emer, leitender Lösungsarchitekt, AWS
- Clark Richey, Hauptarchitekt für Lösungen, AWS
- Elaine Harvey, Geschäftsführerin, Zuverlässigkeitsdienste, AWS
- Jason Barto, leitender Lösungsarchitekt, AWS
- John Formento, Hauptarchitekt für Lösungen, AWS
- Lisi Lewis, leitende Produktmarketing-Managerin, AWS
- Michael Haken, leitender Lösungsarchitekt, AWS
- Neeraj Kumar, Hauptarchitekt für Lösungen, AWS
- Wangechi Doble, Hauptarchitekt für Lösungen, AWS

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	—	06. Oktober 2023

AWS Glossar zu präskriptiven Leitlinien

Im Folgenden finden Sie häufig verwendete Begriffe in Strategien, Leitfäden und Mustern von AWS Prescriptive Guidance. Um Einträge vorzuschlagen, verwenden Sie bitte den Link Feedback geben am Ende des Glossars.

Zahlen

7 Rs

Sieben gängige Migrationsstrategien für die Verlagerung von Anwendungen in die Cloud. Diese Strategien bauen auf den 5 Rs auf, die Gartner 2011 identifiziert hat, und bestehen aus folgenden Elementen:

- Faktorwechsel/Architekturwechsel – Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile cloudnativer Feature nutzen, um Agilität, Leistung und Skalierbarkeit zu verbessern. Dies beinhaltet in der Regel die Portierung des Betriebssystems und der Datenbank. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank auf die Amazon Aurora PostgreSQL-kompatible Edition.
- Plattformwechsel (Lift and Reshape) – Verschieben Sie eine Anwendung in die Cloud und führen Sie ein gewisses Maß an Optimierung ein, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Amazon Relational Database Service (Amazon RDS) für Oracle in der. AWS Cloud
- Neukauf (Drop and Shop) – Wechseln Sie zu einem anderen Produkt, indem Sie typischerweise von einer herkömmlichen Lizenz zu einem SaaS-Modell wechseln. Beispiel: Migrieren Sie Ihr CRM-System (Customer Relationship Management) zu Salesforce.com.
- Hostwechsel (Lift and Shift) – Verschieben Sie eine Anwendung in die Cloud, ohne Änderungen vorzunehmen, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Oracle auf einer EC2-Instanz in der. AWS Cloud
- Verschieben (Lift and Shift auf Hypervisor-Ebene) – Verlagern Sie die Infrastruktur in die Cloud, ohne neue Hardware kaufen, Anwendungen umschreiben oder Ihre bestehenden Abläufe ändern zu müssen. Sie migrieren Server von einer lokalen Plattform zu einem Cloud-Dienst für dieselbe Plattform. Beispiel: Migrieren Sie eine Microsoft Hyper-V Anwendung zu AWS.
- Beibehaltung (Wiederaufgreifen) – Bewahren Sie Anwendungen in Ihrer Quellumgebung auf. Dazu können Anwendungen gehören, die einen umfangreichen Faktorwechsel erfordern und

die Sie auf einen späteren Zeitpunkt verschieben möchten, sowie ältere Anwendungen, die Sie beibehalten möchten, da es keine geschäftliche Rechtfertigung für ihre Migration gibt.

- Außerbetriebnahme – Dekommissionierung oder Entfernung von Anwendungen, die in Ihrer Quellumgebung nicht mehr benötigt werden.

A

ABAC

Siehe attributbasierte Zugriffskontrolle.

abstrahierte Dienste

Weitere Informationen finden Sie unter Managed Services.

ACID

Siehe Atomarität, Konsistenz, Isolierung und Haltbarkeit.

Aktiv-Aktiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden (mithilfe eines bidirektionalen Replikationstools oder dualer Schreibvorgänge) und beide Datenbanken Transaktionen von miteinander verbundenen Anwendungen während der Migration verarbeiten. Diese Methode unterstützt die Migration in kleinen, kontrollierten Batches, anstatt einen einmaligen Cutover zu erfordern. Es ist flexibler, erfordert aber mehr Arbeit als eine aktiv-passive Migration.

Aktiv-Passiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden, aber nur die Quelldatenbank Transaktionen von verbindenden Anwendungen verarbeitet, während Daten in die Zieldatenbank repliziert werden. Die Zieldatenbank akzeptiert während der Migration keine Transaktionen.

Aggregatfunktion

Eine SQL-Funktion, die mit einer Gruppe von Zeilen arbeitet und einen einzelnen Rückgabewert für die Gruppe berechnet. Beispiele für Aggregatfunktionen sind `SUM` und`MAX`.

AI

Siehe künstliche Intelligenz.

AIOps

Siehe Operationen mit künstlicher Intelligenz.

Anonymisierung

Der Prozess des dauerhaften Löschens personenbezogener Daten in einem Datensatz. Anonymisierung kann zum Schutz der Privatsphäre beitragen. Anonymisierte Daten gelten nicht mehr als personenbezogene Daten.

Anti-Muster

Eine häufig verwendete Lösung für ein wiederkehrendes Problem, bei dem die Lösung kontraproduktiv, ineffektiv oder weniger wirksam als eine Alternative ist.

Anwendungssteuerung

Ein Sicherheitsansatz, bei dem nur zugelassene Anwendungen verwendet werden können, um ein System vor Schadsoftware zu schützen.

Anwendungsportfolio

Eine Sammlung detaillierter Informationen zu jeder Anwendung, die von einer Organisation verwendet wird, einschließlich der Kosten für die Erstellung und Wartung der Anwendung und ihres Geschäftswerts. Diese Informationen sind entscheidend für den Prozess der Portfoliofindung und -analyse und hilft bei der Identifizierung und Priorisierung der Anwendungen, die migriert, modernisiert und optimiert werden sollen.

künstliche Intelligenz (KI)

Das Gebiet der Datenverarbeitungswissenschaft, das sich der Nutzung von Computertechnologien zur Ausführung kognitiver Funktionen widmet, die typischerweise mit Menschen in Verbindung gebracht werden, wie Lernen, Problemlösen und Erkennen von Mustern. Weitere Informationen finden Sie unter Was ist künstliche Intelligenz?

Operationen mit künstlicher Intelligenz (AIOps)

Der Prozess des Einsatzes von Techniken des Machine Learning zur Lösung betrieblicher Probleme, zur Reduzierung betrieblicher Zwischenfälle und menschlicher Eingriffe sowie zur Steigerung der Servicequalität. Weitere Informationen zur Verwendung von AIOps in der AWS -Migrationsstrategie finden Sie im Leitfaden zur Betriebsintegration.

Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den

öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

Eine Reihe von Softwareeigenschaften, die die Datenvalidität und betriebliche Zuverlässigkeit einer Datenbank auch bei Fehlern, Stromausfällen oder anderen Problemen gewährleisten.

Attributbasierte Zugriffskontrolle (ABAC)

Die Praxis, detaillierte Berechtigungen auf der Grundlage von Benutzerattributen wie Abteilung, Aufgabenrolle und Teamname zu erstellen. Weitere Informationen finden Sie unter ABAC AWS in der AWS Identity and Access Management (IAM-) Dokumentation.

autoritative Datenquelle

Ein Ort, an dem Sie die primäre Version der Daten speichern, die als die zuverlässigste Informationsquelle angesehen wird. Sie können Daten aus der maßgeblichen Datenquelle an andere Speicherorte kopieren, um die Daten zu verarbeiten oder zu ändern, z. B. zu anonymisieren, zu redigieren oder zu pseudonymisieren.

Availability Zone

Ein bestimmter Standort innerhalb einer AWS-Region , der vor Ausfällen in anderen Availability Zones geschützt ist und kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben Region bietet.

AWS Framework für die Cloud-Einführung (AWS CAF)

Ein Framework mit Richtlinien und bewährten Verfahren, das Unternehmen bei der Entwicklung eines effizienten und effektiven Plans für den erfolgreichen Umstieg auf die Cloud unterstützt. AWS AWS CAF unterteilt die Leitlinien in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Unternehmen, Mitarbeiter, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Aus dieser Perspektive bietet AWS CAF Leitlinien für Personalentwicklung, Schulung und Kommunikation, um das Unternehmen auf eine erfolgreiche Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der AWS -CAF-Webseite und dem AWS -CAF-Whitepaper.

AWS Workload-Qualifizierungsrahmen (AWS WQF)

Ein Tool, das Workloads bei der Datenbankmigration bewertet, Migrationsstrategien empfiehlt und Arbeitsschätzungen bereitstellt. AWS WQF ist in () enthalten. AWS Schema Conversion Tool AWS SCT Es analysiert Datenbankschemas und Codeobjekte, Anwendungscode, Abhängigkeiten und Leistungsmerkmale und stellt Bewertungsberichte bereit.

B

schlechter Bot

Ein Bot, der Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen soll.

BCP

Siehe Planung der Geschäftskontinuität.

Verhaltensdiagramm

Eine einheitliche, interaktive Ansicht des Ressourcenverhaltens und der Interaktionen im Laufe der Zeit. Sie können ein Verhaltensdiagramm mit Amazon Detective verwenden, um fehlgeschlagene Anmeldeversuche, verdächtige API-Aufrufe und ähnliche Vorgänge zu untersuchen. Weitere Informationen finden Sie unter Daten in einem Verhaltensdiagramm in der Detective-Dokumentation.

Big-Endian-System

Ein System, welches das höchstwertige Byte zuerst speichert. Siehe auch Endianness.

Binäre Klassifikation

Ein Prozess, der ein binäres Ergebnis vorhersagt (eine von zwei möglichen Klassen). Beispielsweise könnte Ihr ML-Modell möglicherweise Probleme wie „Handelt es sich bei dieser E-Mail um Spam oder nicht?“ vorhersagen müssen oder „Ist dieses Produkt ein Buch oder ein Auto?“

Bloom-Filter

Eine probabilistische, speichereffiziente Datenstruktur, mit der getestet wird, ob ein Element Teil einer Menge ist.

Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue

Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

Bot

Eine Softwareanwendung, die automatisierte Aufgaben über das Internet ausführt und menschliche Aktivitäten oder Interaktionen simuliert. Manche Bots sind nützlich oder nützlich, wie z. B. Webcrawler, die Informationen im Internet indexieren. Einige andere Bots, die als bösartige Bots bezeichnet werden, sollen Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen.

Botnetz

Netzwerke von Bots, die mit Malware infiziert sind und unter der Kontrolle einer einzigen Partei stehen, die als Bot-Herder oder Bot-Operator bezeichnet wird. Botnetze sind der bekannteste Mechanismus zur Skalierung von Bots und ihrer Wirkung.

branch

Ein containerisierter Bereich eines Code-Repositorys. Der erste Zweig, der in einem Repository erstellt wurde, ist der Hauptzweig. Sie können einen neuen Zweig aus einem vorhandenen Zweig erstellen und dann Feature entwickeln oder Fehler in dem neuen Zweig beheben. Ein Zweig, den Sie erstellen, um ein Feature zu erstellen, wird allgemein als Feature-Zweig bezeichnet. Wenn das Feature zur Veröffentlichung bereit ist, führen Sie den Feature-Zweig wieder mit dem Hauptzweig zusammen. Weitere Informationen finden Sie unter Über Branches (GitHub Dokumentation).

Zugang durch Glasbruch

Unter außergewöhnlichen Umständen und im Rahmen eines genehmigten Verfahrens ist dies eine schnelle Methode für einen Benutzer, auf einen Bereich zuzugreifen AWS-Konto , für den er normalerweise keine Zugriffsrechte besitzt. Weitere Informationen finden Sie unter dem Indikator Implementation break-glass procedures in den AWS Well-Architected-Leitlinien.

Brownfield-Strategie

Die bestehende Infrastruktur in Ihrer Umgebung. Wenn Sie eine Brownfield-Strategie für eine Systemarchitektur anwenden, richten Sie sich bei der Gestaltung der Architektur nach den Einschränkungen der aktuellen Systeme und Infrastruktur. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Puffer-Cache

Der Speicherbereich, in dem die am häufigsten abgerufenen Daten gespeichert werden.

Geschäftsfähigkeit

Was ein Unternehmen tut, um Wert zu generieren (z. B. Vertrieb, Kundenservice oder Marketing). Microservices-Architekturen und Entwicklungsentscheidungen können von den Geschäftskapazitäten beeinflusst werden. Weitere Informationen finden Sie im Abschnitt Organisiert nach Geschäftskapazitäten des Whitepapers Ausführen von containerisierten Microservices in AWS.

Planung der Geschäftskontinuität (BCP)

Ein Plan, der die potenziellen Auswirkungen eines störenden Ereignisses, wie z. B. einer groß angelegten Migration, auf den Betrieb berücksichtigt und es einem Unternehmen ermöglicht, den Betrieb schnell wieder aufzunehmen.

C

CAF

Weitere Informationen finden Sie unter Framework für die AWS Cloud-Einführung.

Bereitstellung auf Kanaren

Die langsame und schrittweise Veröffentlichung einer Version für Endbenutzer. Wenn Sie sich sicher sind, stellen Sie die neue Version bereit und ersetzen die aktuelle Version vollständig.

CCoE

Weitere Informationen finden Sie im Cloud Center of Excellence.

CDC

Siehe Erfassung von Änderungsdaten.

Erfassung von Datenänderungen (CDC)

Der Prozess der Nachverfolgung von Änderungen an einer Datenquelle, z. B. einer Datenbanktabelle, und der Aufzeichnung von Metadaten zu der Änderung. Sie können CDC für verschiedene Zwecke verwenden, z. B. für die Prüfung oder Replikation von Änderungen in einem Zielsystem, um die Synchronisation aufrechtzuerhalten.

Chaos-Technik

Absichtliches Einführen von Ausfällen oder Störungsereignissen, um die Widerstandsfähigkeit eines Systems zu testen. Sie können AWS Fault Injection Service (AWS FIS) verwenden, um Experimente durchzuführen, die Ihre AWS Workloads stress, und deren Reaktion zu bewerten.

CI/CD

Siehe Continuous Integration und Continuous Delivery.

Klassifizierung

Ein Kategorisierungsprozess, der bei der Erstellung von Vorhersagen hilft. ML-Modelle für Klassifikationsprobleme sagen einen diskreten Wert voraus. Diskrete Werte unterscheiden sich immer voneinander. Beispielsweise muss ein Modell möglicherweise auswerten, ob auf einem Bild ein Auto zu sehen ist oder nicht.

clientseitige Verschlüsselung

Lokale Verschlüsselung von Daten, bevor das Ziel sie AWS -Service empfängt.

Cloud-Kompetenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte Transformationen. Weitere Informationen finden Sie in den CCoE-Beiträgen im AWS Cloud Enterprise Strategy Blog.

Cloud Computing

Die Cloud-Technologie, die typischerweise für die Ferndatenspeicherung und das IoT-Gerätemanagement verwendet wird. Cloud Computing ist häufig mit Edge-Computing-Technologie verbunden.

Cloud-Betriebsmodell

In einer IT-Organisation das Betriebsmodell, das zum Aufbau, zur Weiterentwicklung und Optimierung einer oder mehrerer Cloud-Umgebungen verwendet wird. Weitere Informationen finden Sie unter Aufbau Ihres Cloud-Betriebsmodells.

Phasen der Einführung der Cloud

Die vier Phasen, die Unternehmen bei der Migration in der Regel durchlaufen AWS Cloud:

- Projekt – Durchführung einiger Cloud-bezogener Projekte zu Machbarkeitsnachweisen und zu Lernzwecken
- Fundament – Grundlegende Investitionen tätigen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer Landing Zone, Definition eines CCoE, Einrichtung eines Betriebsmodells)
- Migration – Migrieren einzelner Anwendungen

- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

Diese Phasen wurden von Stephen Orban im Blogbeitrag The Journey Toward Cloud-First & the Stages of Adoption im AWS Cloud Enterprise Strategy-Blog definiert. Informationen darüber, wie sie mit der AWS Migrationsstrategie zusammenhängen, finden Sie im Leitfaden zur Vorbereitung der Migration.

CMDB

Siehe Datenbank für das Konfigurationsmanagement.

Code-Repository

Ein Ort, an dem Quellcode und andere Komponenten wie Dokumentation, Beispiele und Skripts gespeichert und im Rahmen von Versionskontrollprozessen aktualisiert werden. Zu den gängigen Cloud-Repositorys gehören GitHub oder AWS CodeCommit. Jede Version des Codes wird Zweig genannt. In einer Microservice-Struktur ist jedes Repository einer einzelnen Funktionalität gewidmet. Eine einzelne CI/CD-Pipeline kann mehrere Repositorien verwenden.

Kalter Cache

Ein Puffer-Cache, der leer oder nicht gut gefüllt ist oder veraltete oder irrelevante Daten enthält. Dies beeinträchtigt die Leistung, da die Datenbank-Instance aus dem Hauptspeicher oder der Festplatte lesen muss, was langsamer ist als das Lesen aus dem Puffercache.

Kalte Daten

Daten, auf die selten zugegriffen wird und die in der Regel historisch sind. Bei der Abfrage dieser Art von Daten sind langsame Abfragen in der Regel akzeptabel. Durch die Verlagerung dieser Daten auf leistungsschwächere und kostengünstigere Speicherstufen oder -klassen können Kosten gesenkt werden.

Computer Vision (CV)

Ein Bereich der KI, der maschinelles Lernen nutzt, um Informationen aus visuellen Formaten wie digitalen Bildern und Videos zu analysieren und zu extrahieren. AWS Panorama Bietet beispielsweise Geräte an, die CV zu lokalen Kameranetzwerken hinzufügen, und Amazon SageMaker stellt Bildverarbeitungsalgorithmen für CV bereit.

Drift in der Konfiguration

Bei einer Arbeitslast eine Änderung der Konfiguration gegenüber dem erwarteten Zustand. Dies kann dazu führen, dass der Workload nicht mehr richtlinienkonform wird, und zwar in der Regel schrittweise und unbeabsichtigt.

Verwaltung der Datenbankkonfiguration (CMDB)

Ein Repository, das Informationen über eine Datenbank und ihre IT-Umgebung speichert und verwaltet, inklusive Hardware- und Softwarekomponenten und deren Konfigurationen. In der Regel verwenden Sie Daten aus einer CMDB in der Phase der Portfolioerkennung und -analyse der Migration.

Konformitätspaket

Eine Sammlung von AWS Config Regeln und Abhilfemaßnahmen, die Sie zusammenstellen können, um Ihre Konformitäts- und Sicherheitsprüfungen individuell anzupassen. Mithilfe einer YAML-Vorlage können Sie ein Conformance Pack als einzelne Entität in einer AWS-Konto AND-Region oder unternehmensweit bereitstellen. Weitere Informationen finden Sie in der Dokumentation unter Conformance Packs. AWS Config

Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

Der Prozess der Automatisierung der Quell-, Build-, Test-, Staging- und Produktionsphasen des Softwareveröffentlichungsprozesses. CI/CD wird allgemein als Pipeline beschrieben. CI/CD kann Ihnen helfen, Prozesse zu automatisieren, die Produktivität zu steigern, die Codequalität zu verbessern und schneller zu liefern. Weitere Informationen finden Sie unter Vorteile der kontinuierlichen Auslieferung. CD kann auch für kontinuierliche Bereitstellung stehen. Weitere Informationen finden Sie unterKontinuierliche Auslieferung im Vergleich zu kontinuierlicher Bereitstellung.

CV

Siehe Computer Vision.

D

Daten im Ruhezustand

Daten, die in Ihrem Netzwerk stationär sind, z. B. Daten, die sich im Speicher befinden.

Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil

der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter Datenklassifizierung.

Datendrift

Eine signifikante Abweichung zwischen den Produktionsdaten und den Daten, die zum Trainieren eines ML-Modells verwendet wurden, oder eine signifikante Änderung der Eingabedaten im Laufe der Zeit. Datendrift kann die Gesamtqualität, Genauigkeit und Fairness von ML-Modellvorhersagen beeinträchtigen.

Daten während der Übertragung

Daten, die sich aktiv durch Ihr Netzwerk bewegen, z. B. zwischen Netzwerkressourcen.

Datennetz

Ein architektonisches Framework, das verteilte, dezentrale Dateneigentum mit zentraler Verwaltung und Steuerung ermöglicht.

Datenminimierung

Das Prinzip, nur die Daten zu sammeln und zu verarbeiten, die unbedingt erforderlich sind. Durch Datenminimierung im AWS Cloud können Datenschutzrisiken, Kosten und der CO2-Fußabdruck Ihrer Analysen reduziert werden.

Datenperimeter

Eine Reihe präventiver Schutzmaßnahmen in Ihrer AWS Umgebung, die sicherstellen, dass nur vertrauenswürdige Identitäten auf vertrauenswürdige Ressourcen von erwarteten Netzwerken zugreifen. Weitere Informationen finden Sie unter Aufbau eines Datenperimeters auf. AWS

Vorverarbeitung der Daten

Rohdaten in ein Format umzuwandeln, das von Ihrem ML-Modell problemlos verarbeitet werden kann. Die Vorverarbeitung von Daten kann bedeuten, dass bestimmte Spalten oder Zeilen entfernt und fehlende, inkonsistente oder doppelte Werte behoben werden.

Herkunft der Daten

Der Prozess der Nachverfolgung des Ursprungs und der Geschichte von Daten während ihres gesamten Lebenszyklus, z. B. wie die Daten generiert, übertragen und gespeichert wurden.

betroffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

Ein Datenverwaltungssystem, das Business Intelligence wie Analysen unterstützt. Data Warehouses enthalten in der Regel große Mengen an historischen Daten und werden in der Regel für Abfragen und Analysen verwendet.

Datenbankdefinitionssprache (DDL)

Anweisungen oder Befehle zum Erstellen oder Ändern der Struktur von Tabellen und Objekten in einer Datenbank.

Datenbankmanipulationssprache (DML)

Anweisungen oder Befehle zum Ändern (Einfügen, Aktualisieren und Löschen) von Informationen in einer Datenbank.

DDL

Siehe Datenbankdefinitionssprache.

Deep-Ensemble

Mehrere Deep-Learning-Modelle zur Vorhersage kombinieren. Sie können Deep-Ensembles verwenden, um eine genauere Vorhersage zu erhalten oder um die Unsicherheit von Vorhersagen abzuschätzen.

Deep Learning

Ein ML-Teilbereich, der mehrere Schichten künstlicher neuronaler Netzwerke verwendet, um die Zuordnung zwischen Eingabedaten und Zielvariablen von Interesse zu ermitteln.

defense-in-depth

Ein Ansatz zur Informationssicherheit, bei dem eine Reihe von Sicherheitsmechanismen und -kontrollen sorgfältig in einem Computernetzwerk verteilt werden, um die Vertraulichkeit, Integrität und Verfügbarkeit des Netzwerks und der darin enthaltenen Daten zu schützen. Wenn Sie diese Strategie anwenden AWS, fügen Sie mehrere Steuerelemente auf verschiedenen Ebenen der AWS Organizations Struktur hinzu, um die Ressourcen zu schützen. Ein defense-in-depth Ansatz könnte beispielsweise Multi-Faktor-Authentifizierung, Netzwerksegmentierung und Verschlüsselung kombinieren.

delegierter Administrator

In AWS Organizations kann ein kompatibler Dienst ein AWS Mitgliedskonto registrieren, um die Konten der Organisation und die Berechtigungen für diesen Dienst zu verwalten. Dieses Konto

wird als delegierter Administrator für diesen Service bezeichnet. Weitere Informationen und eine Liste kompatibler Services finden Sie unter Services, die mit AWS Organizations funktionieren in der AWS Organizations -Dokumentation.

Bereitstellung

Der Prozess, bei dem eine Anwendung, neue Feature oder Codekorrekturen in der Zielumgebung verfügbar gemacht werden. Die Bereitstellung umfasst das Implementieren von Änderungen an einer Codebasis und das anschließende Erstellen und Ausführen dieser Codebasis in den Anwendungsumgebungen.

Entwicklungsumgebung

Siehe Umgebung.

Detektivische Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, ein Ereignis zu erkennen, zu protokollieren und zu warnen, nachdem ein Ereignis eingetreten ist. Diese Kontrollen stellen eine zweite Verteidigungslinie dar und warnen Sie vor Sicherheitsereignissen, bei denen die vorhandenen präventiven Kontrollen umgangen wurden. Weitere Informationen finden Sie unter Detektivische Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Abbildung des Wertstroms in der Entwicklung (DVSM)

Ein Prozess zur Identifizierung und Priorisierung von Einschränkungen, die sich negativ auf Geschwindigkeit und Qualität im Lebenszyklus der Softwareentwicklung auswirken. DVSM erweitert den Prozess der Wertstromanalyse, der ursprünglich für Lean-Manufacturing-Praktiken konzipiert wurde. Es konzentriert sich auf die Schritte und Teams, die erforderlich sind, um durch den Softwareentwicklungsprozess Mehrwert zu schaffen und zu steigern.

digitaler Zwilling

Eine virtuelle Darstellung eines realen Systems, z. B. eines Gebäudes, einer Fabrik, einer Industrieanlage oder einer Produktionslinie. Digitale Zwillinge unterstützen vorausschauende Wartung, Fernüberwachung und Produktionsoptimierung.

Maßtabelle

In einem Sternschema eine kleinere Tabelle, die Datenattribute zu quantitativen Daten in einer Faktentabelle enthält. Bei Attributen von Dimensionstabellen handelt es sich in der Regel um Textfelder oder diskrete Zahlen, die sich wie Text verhalten. Diese Attribute werden häufig zum Einschränken von Abfragen, zum Filtern und zur Kennzeichnung von Ergebnismengen verwendet.

Katastrophe

Ein Ereignis, das verhindert, dass ein Workload oder ein System seine Geschäftsziele an seinem primären Einsatzort erfüllt. Diese Ereignisse können Naturkatastrophen, technische Ausfälle oder das Ergebnis menschlichen Handelns sein, z. B. unbeabsichtigte Fehlkonfigurationen oder Malware-Angriffe.

Notfallwiederherstellung (DR)

Die Strategie und der Prozess, die Sie zur Minimierung von Ausfallzeiten und Datenverlusten aufgrund einer Katastrophe anwenden. Weitere Informationen finden Sie unter Disaster Recovery von Workloads unter AWS: Wiederherstellung in der Cloud im AWS Well-Architected Framework.

DML

Siehe Sprache zur Datenbankmanipulation.

Domainorientiertes Design

Ein Ansatz zur Entwicklung eines komplexen Softwaresystems, bei dem seine Komponenten mit sich entwickelnden Domains oder Kerngeschäftszielen verknüpft werden, denen jede Komponente dient. Dieses Konzept wurde von Eric Evans in seinem Buch Domaingesteuertes Design: Bewältigen der Komplexität im Herzen der Software (Boston: Addison-Wesley Professional, 2003) vorgestellt. Informationen darüber, wie Sie domaingesteuertes Design mit dem Strangler-Fig-Muster verwenden können, finden Sie unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

DR

Siehe Disaster Recovery.

Erkennung von Driften

Verfolgung von Abweichungen von einer Basiskonfiguration Sie können es beispielsweise verwenden, AWS CloudFormation um Abweichungen bei den Systemressourcen zu erkennen, oder Sie können AWS Control Tower damit Änderungen in Ihrer landing zone erkennen, die sich auf die Einhaltung von Governance-Anforderungen auswirken könnten.

DVSM

Siehe Abbildung des Wertstroms in der Entwicklung.

E

EDA

Siehe explorative Datenanalyse.

Edge-Computing

Die Technologie, die die Rechenleistung für intelligente Geräte an den Rändern eines IoT-Netzwerks erhöht. Im Vergleich zu Cloud Computing kann Edge Computing die Kommunikationslatenz reduzieren und die Reaktionszeit verbessern.

Verschlüsselung

Ein Rechenprozess, der Klartextdaten, die für Menschen lesbar sind, in Chiffretext umwandelt.

Verschlüsselungsschlüssel

Eine kryptografische Zeichenfolge aus zufälligen Bits, die von einem Verschlüsselungsalgorithmus generiert wird. Schlüssel können unterschiedlich lang sein, und jeder Schlüssel ist so konzipiert, dass er unvorhersehbar und einzigartig ist.

Endianismus

Die Reihenfolge, in der Bytes im Computerspeicher gespeichert werden. Big-Endian-Systeme speichern das höchstwertige Byte zuerst. Little-Endian-Systeme speichern das niedrigwertigste Byte zuerst.

Endpunkt

Siehe Service-Endpunkt.

Endpunkt-Services

Ein Service, den Sie in einer Virtual Private Cloud (VPC) hosten können, um ihn mit anderen Benutzern zu teilen. Sie können einen Endpunktdienst mit anderen AWS-Konten oder AWS Identity and Access Management (IAM AWS PrivateLink -) Prinzipalen erstellen und diesen Berechtigungen gewähren. Diese Konten oder Prinzipale können sich privat mit Ihrem Endpunktservice verbinden, indem sie Schnittstellen-VPC-Endpunkte erstellen. Weitere Informationen finden Sie unter Einen Endpunkt-Service erstellen in der Amazon Virtual Private Cloud (Amazon VPC)-Dokumentation.

Unternehmensressourcenplanung (ERP)

Ein System, das wichtige Geschäftsprozesse (wie Buchhaltung, MES und Projektmanagement) für ein Unternehmen automatisiert und verwaltet.

Envelope-Verschlüsselung

Der Prozess der Verschlüsselung eines Verschlüsselungsschlüssels mit einem anderen Verschlüsselungsschlüssel. Weitere Informationen finden Sie unter Envelope-Verschlüsselung in der AWS Key Management Service (AWS KMS) -Dokumentation.

Umgebung

Eine Instance einer laufenden Anwendung. Die folgenden Arten von Umgebungen sind beim Cloud-Computing üblich:

- Entwicklungsumgebung – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.
- Niedrigere Umgebungen – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.
- Produktionsumgebung – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD-Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

Epics

In der agilen Methodik sind dies funktionale Kategorien, die Ihnen helfen, Ihre Arbeit zu organisieren und zu priorisieren. Epics bieten eine allgemeine Beschreibung der Anforderungen und Implementierungsaufgaben. Zu den Sicherheitsthemen AWS von CAF gehören beispielsweise Identitäts- und Zugriffsmanagement, Detektivkontrollen, Infrastruktursicherheit, Datenschutz und Reaktion auf Vorfälle. Weitere Informationen zu Epics in der AWS -Migrationsstrategie finden Sie im Leitfaden zur Programm-Implementierung.

ERP

Siehe Enterprise Resource Planning.

Explorative Datenanalyse (EDA)

Der Prozess der Analyse eines Datensatzes, um seine Hauptmerkmale zu verstehen. Sie sammeln oder aggregieren Daten und führen dann erste Untersuchungen durch, um Muster zu

finden, Anomalien zu erkennen und Annahmen zu überprüfen. EDA wird durchgeführt, indem zusammenfassende Statistiken berechnet und Datenvisualisierungen erstellt werden.

F

Faktentabelle

Die zentrale Tabelle in einem Sternschema. Sie speichert quantitative Daten über den Geschäftsbetrieb. In der Regel enthält eine Faktentabelle zwei Arten von Spalten: Spalten, die Kennzahlen enthalten, und Spalten, die einen Fremdschlüssel für eine Dimensionstabelle enthalten.

schnell scheitern

Eine Philosophie, die häufige und inkrementelle Tests verwendet, um den Entwicklungslebenszyklus zu verkürzen. Dies ist ein wichtiger Bestandteil eines agilen Ansatzes.

Grenze zur Fehlerisolierung

Dabei handelt es sich um eine Grenze AWS Cloud, z. B. eine Availability Zone AWS-Region, eine Steuerungsebene oder eine Datenebene, die die Auswirkungen eines Fehlers begrenzt und die Widerstandsfähigkeit von Workloads verbessert. Weitere Informationen finden Sie unter Grenzen zur AWS Fehlerisolierung.

Feature-Zweig

Siehe Zweig.

Features

Die Eingabedaten, die Sie verwenden, um eine Vorhersage zu treffen. In einem Fertigungskontext könnten Feature beispielsweise Bilder sein, die regelmäßig von der Fertigungslinie aus aufgenommen werden.

Bedeutung der Feature

Wie wichtig ein Feature für die Vorhersagen eines Modells ist. Dies wird in der Regel als numerischer Wert ausgedrückt, der mit verschiedenen Techniken wie Shapley Additive Explanations (SHAP) und integrierten Gradienten berechnet werden kann. Weitere Informationen finden Sie unter Interpretierbarkeit von Modellen für maschinelles Lernen mit:AWS.

Featuretransformation

Daten für den ML-Prozess optimieren, einschließlich der Anreicherung von Daten mit zusätzlichen Quellen, der Skalierung von Werten oder der Extraktion mehrerer Informationssätze aus einem einzigen Datenfeld. Das ermöglicht dem ML-Modell, von den Daten profitieren. Wenn Sie beispielsweise das Datum „27.05.2021 00:15:37“ in „2021“, „Mai“, „Donnerstag“ und „15“ aufschlüsseln, können Sie dem Lernalgorithmus helfen, nuancierte Muster zu erlernen, die mit verschiedenen Datenkomponenten verknüpft sind.

FGAC

Weitere Informationen finden Sie unter detaillierter Zugriffskontrolle.

Feinkörnige Zugriffskontrolle (FGAC)

Die Verwendung mehrerer Bedingungen, um eine Zugriffsanfrage zuzulassen oder abzulehnen.

Flash-Cut-Migration

Eine Datenbankmigrationsmethode, bei der eine kontinuierliche Datenreplikation durch Erfassung von Änderungsdaten verwendet wird, um Daten in kürzester Zeit zu migrieren, anstatt einen schrittweisen Ansatz zu verwenden. Ziel ist es, Ausfallzeiten auf ein Minimum zu beschränken.

G

Geoblocking

Siehe geografische Einschränkungen.

Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden, um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte. CloudFront

Gitflow-Workflow

Ein Ansatz, bei dem niedrigere und höhere Umgebungen unterschiedliche Zweige in einem Quellcode-Repository verwenden. Der Gitflow-Workflow gilt als veraltet, und der Trunk-basierte Workflow ist der moderne, bevorzugte Ansatz.

Greenfield-Strategie

Das Fehlen vorhandener Infrastruktur in einer neuen Umgebung. Bei der Einführung einer Neuausrichtung einer Systemarchitektur können Sie alle neuen Technologien ohne Einschränkung der Kompatibilität mit der vorhandenen Infrastruktur auswählen, auch bekannt als Brownfield. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Integritätsschutz

Eine allgemeine Regel, die dabei hilft, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Organisationseinheiten (OUs) zu regeln. Präventiver Integritätsschutz setzt Richtlinien durch, um die Einhaltung von Standards zu gewährleisten. Sie werden mithilfe von Service-Kontrollrichtlinien und IAM-Berechtigungsgrenzen implementiert. Detektivischer Integritätsschutz erkennt Richtlinienverstöße und Compliance-Probleme und generiert Warnmeldungen zur Abhilfe. Sie werden mithilfe von AWS Config, AWS Security Hub, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector und benutzerdefinierten AWS Lambda Prüfungen implementiert.

H

HEKTAR

Siehe Hochverfügbarkeit.

Heterogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank in eine Zieldatenbank, die eine andere Datenbank-Engine verwendet (z. B. Oracle zu Amazon Aurora). Eine heterogene Migration ist in der Regel Teil einer Neuarchitektur, und die Konvertierung des Schemas kann eine komplexe Aufgabe sein. AWS bietet AWS SCT, welches bei Schemakonvertierungen hilft.

hohe Verfügbarkeit (HA)

Die Fähigkeit eines Workloads, im Falle von Herausforderungen oder Katastrophen kontinuierlich und ohne Eingreifen zu arbeiten. HA-Systeme sind so konzipiert, dass sie automatisch ein Failover durchführen, gleichbleibend hohe Leistung bieten und unterschiedliche Lasten und Ausfälle mit minimalen Leistungseinbußen bewältigen.

historische Modernisierung

Ein Ansatz zur Modernisierung und Aufrüstung von Betriebstechnologiesystemen (OT), um den Bedürfnissen der Fertigungsindustrie besser gerecht zu werden. Ein Historian ist eine Art von Datenbank, die verwendet wird, um Daten aus verschiedenen Quellen in einer Fabrik zu sammeln und zu speichern.

Homogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank zu einer Zieldatenbank, die dieselbe Datenbank-Engine verwendet (z. B. Microsoft SQL Server zu Amazon RDS für SQL Server). Eine homogene Migration ist in der Regel Teil eines Hostwechsels oder eines Plattformwechsels. Sie können native Datenbankserviceprogramme verwenden, um das Schema zu migrieren.

heiße Daten

Daten, auf die häufig zugegriffen wird, z. B. Echtzeitdaten oder aktuelle Translationsdaten. Für diese Daten ist in der Regel eine leistungsstarke Speicherebene oder -klasse erforderlich, um schnelle Abfrageantworten zu ermöglichen.

Hotfix

Eine dringende Lösung für ein kritisches Problem in einer Produktionsumgebung. Aufgrund seiner Dringlichkeit wird ein Hotfix normalerweise außerhalb des typischen DevOps Release-Workflows erstellt.

Hypercare-Phase

Unmittelbar nach dem Cutover, der Zeitraum, in dem ein Migrationsteam die migrierten Anwendungen in der Cloud verwaltet und überwacht, um etwaige Probleme zu beheben. In der Regel dauert dieser Zeitraum 1–4 Tage. Am Ende der Hypercare-Phase überträgt das Migrationsteam in der Regel die Verantwortung für die Anwendungen an das Cloud-Betriebsteam.

I

IaC

Sehen Sie sich Infrastruktur als Code an.

Identitätsbasierte Richtlinie

Eine Richtlinie, die einem oder mehreren IAM-Prinzipalen zugeordnet ist und deren Berechtigungen innerhalb der AWS Cloud Umgebung definiert.

Leerlaufanwendung

Eine Anwendung mit einer durchschnittlichen CPU- und Arbeitsspeicherauslastung zwischen 5 und 20 Prozent über einen Zeitraum von 90 Tagen. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen oder sie On-Premises beizubehalten.

IIoT

Siehe Industrielles Internet der Dinge.

unveränderliche Infrastruktur

Ein Modell, das eine neue Infrastruktur für Produktionsworkloads bereitstellt, anstatt die bestehende Infrastruktur zu aktualisieren, zu patchen oder zu modifizieren. Unveränderliche Infrastrukturen sind von Natur aus konsistenter, zuverlässiger und vorhersehbarer als veränderliche Infrastrukturen. Weitere Informationen finden Sie in der Best Practice Deploy using immutable infrastructure im AWS Well-Architected Framework.

Eingehende (ingress) VPC

In einer Architektur AWS mit mehreren Konten ist dies eine VPC, die Netzwerkverbindungen von außerhalb einer Anwendung akzeptiert, überprüft und weiterleitet. Die AWS -Referenzarchitektur für die Sicherheit empfiehlt, Ihr Netzwerkkonto mit eingehenden und ausgehenden VPCs und Inspektions-VPCs einzurichten, um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet zu schützen.

Inkrementelle Migration

Eine Cutover-Strategie, bei der Sie Ihre Anwendung in kleinen Teilen migrieren, anstatt eine einziges vollständiges Cutover durchzuführen. Beispielsweise könnten Sie zunächst nur einige Microservices oder Benutzer auf das neue System umstellen. Nachdem Sie sich vergewissert haben, dass alles ordnungsgemäß funktioniert, können Sie weitere Microservices oder Benutzer schrittweise verschieben, bis Sie Ihr Legacy-System außer Betrieb nehmen können. Diese Strategie reduziert die mit großen Migrationen verbundenen Risiken.

Industrie 4.0

Ein Begriff, der 2016 von Klaus Schwab eingeführt wurde und sich auf die Modernisierung von Fertigungsprozessen durch Fortschritte in den Bereichen Konnektivität, Echtzeitdaten, Automatisierung, Analytik und KI/ML bezieht.

Infrastruktur

Alle Ressourcen und Komponenten, die in der Umgebung einer Anwendung enthalten sind.

Infrastructure as Code (IaC)

Der Prozess der Bereitstellung und Verwaltung der Infrastruktur einer Anwendung mithilfe einer Reihe von Konfigurationsdateien. IaC soll Ihnen helfen, das Infrastrukturmanagement zu zentralisieren, Ressourcen zu standardisieren und schnell zu skalieren, sodass neue Umgebungen wiederholbar, zuverlässig und konsistent sind.

Industrielles Internet der Dinge (IIoT)

Einsatz von mit dem Internet verbundenen Sensoren und Geräten in Industriesektoren wie Fertigung, Energie, Automobilindustrie, Gesundheitswesen, Biowissenschaften und Landwirtschaft. Mehr Informationen finden Sie unter Aufbau einer digitalen Transformationsstrategie für das industrielle Internet der Dinge (IIoT).

Inspektions-VPC

In einer Architektur AWS mit mehreren Konten eine zentralisierte VPC, die Inspektionen des Netzwerkverkehrs zwischen VPCs (in derselben oder unterschiedlichen AWS-Regionen), dem Internet und lokalen Netzwerken verwaltet. Die AWS -Referenzarchitektur für die Sicherheit empfiehlt, Ihr Netzwerkkonto mit eingehenden und ausgehenden VPCs und Inspektions-VPCs einzurichten, um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet zu schützen.

Internet of Things (IoT)

Das Netzwerk verbundener physischer Objekte mit eingebetteten Sensoren oder Prozessoren, das über das Internet oder über ein lokales Kommunikationsnetzwerk mit anderen Geräten und Systemen kommuniziert. Weitere Informationen finden Sie unter Was ist IoT?

Interpretierbarkeit

Ein Merkmal eines Modells für Machine Learning, das beschreibt, inwieweit ein Mensch verstehen kann, wie die Vorhersagen des Modells von seinen Eingaben abhängen. Weitere Informationen finden Sie unter Interpretierbarkeit von Modellen für Machine Learning mit AWS.

IoT

Siehe Internet der Dinge.

IT information library (ITIL, IT-Informationsbibliothek)

Eine Reihe von bewährten Methoden für die Bereitstellung von IT-Services und die Abstimmung dieser Services auf die Geschäftsanforderungen. ITIL bietet die Grundlage für ITSM.

T service management (ITSM, IT-Servicemanagement)

Aktivitäten im Zusammenhang mit der Gestaltung, Implementierung, Verwaltung und Unterstützung von IT-Services für eine Organisation. Informationen zur Integration von Cloud-Vorgängen mit ITSM-Tools finden Sie im Leitfaden zur Betriebsintegration.

BIS

Weitere Informationen finden Sie in der IT-Informationsbibliothek.

ITSM

Siehe IT-Servicemanagement.

L

Labelbasierte Zugangskontrolle (LBAC)

Eine Implementierung der Mandatory Access Control (MAC), bei der den Benutzern und den Daten selbst jeweils explizit ein Sicherheitslabelwert zugewiesen wird. Die Schnittmenge zwischen der Benutzersicherheitsbeschriftung und der Datensicherheitsbeschriftung bestimmt, welche Zeilen und Spalten für den Benutzer sichtbar sind.

Landing Zone

Eine landing zone ist eine gut strukturierte AWS Umgebung mit mehreren Konten, die skalierbar und sicher ist. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und Infrastrukturumgebung starten und bereitstellen können. Weitere Informationen zu Landing Zones finden Sie unter Einrichtung einer sicheren und skalierbaren AWS -Umgebung mit mehreren Konten..

Große Migration

Eine Migration von 300 oder mehr Servern.

SCHWARZ

Weitere Informationen finden Sie unter Label-basierte Zugriffskontrolle.

Geringste Berechtigung

Die bewährte Sicherheitsmethode, bei der nur die für die Durchführung einer Aufgabe erforderlichen Mindestberechtigungen erteilt werden. Weitere Informationen finden Sie unter Geringste Berechtigungen anwenden in der IAM-Dokumentation.

Lift and Shift

Siehe 7 Rs.

Little-Endian-System

Ein System, welches das niedrigwertigste Byte zuerst speichert. Siehe auch Endianness.

Niedrigere Umgebungen

Siehe Umwelt.

M

Machine Learning (ML)

Eine Art künstlicher Intelligenz, die Algorithmen und Techniken zur Mustererkennung und zum Lernen verwendet. ML analysiert aufgezeichnete Daten, wie z. B. Daten aus dem Internet der Dinge (IoT), und lernt daraus, um ein statistisches Modell auf der Grundlage von Mustern zu erstellen. Weitere Informationen finden Sie unter Machine Learning.

Hauptzweig

Siehe Filiale.

Malware

Software, die entwickelt wurde, um die Computersicherheit oder den Datenschutz zu gefährden. Malware kann Computersysteme stören, vertrauliche Informationen durchsickern lassen oder sich unbefugten Zugriff verschaffen. Beispiele für Malware sind Viren, Würmer, Ransomware, Trojaner, Spyware und Keylogger.

verwaltete Dienste

AWS -Services für die die Infrastrukturebene, das Betriebssystem und die Plattformen AWS betrieben werden, und Sie greifen auf die Endgeräte zu, um Daten zu speichern und abzurufen. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB sind Beispiele für Managed Services. Diese werden auch als abstrakte Dienste bezeichnet.

Manufacturing Execution System (MES)

Ein Softwaresystem zur Nachverfolgung, Überwachung, Dokumentation und Steuerung von Produktionsprozessen, bei denen Rohstoffe in der Fertigung zu fertigen Produkten umgewandelt werden.

MAP

Siehe Migration Acceleration Program.

Mechanismus

Ein vollständiger Prozess, bei dem Sie ein Tool erstellen, die Akzeptanz des Tools vorantreiben und anschließend die Ergebnisse überprüfen, um Anpassungen vorzunehmen. Ein Mechanismus ist ein Zyklus, der sich im Laufe seiner Tätigkeit selbst verstärkt und verbessert. Weitere Informationen finden Sie unter Aufbau von Mechanismen im AWS Well-Architected Framework.

Mitgliedskonto

Alle AWS-Konten außer dem Verwaltungskonto, die Teil einer Organisation in sind. AWS Organizations Ein Konto kann jeweils nur einer Organisation angehören.

DURCHEINANDER

Siehe Manufacturing Execution System.

Message Queuing-Telemetrietransport (MQTT)

Ein leichtes machine-to-machine (M2M) -Kommunikationsprotokoll, das auf dem Publish/Subscribe-Muster für IoT-Geräte mit beschränkten Ressourcen basiert.

Microservice

Ein kleiner, unabhängiger Service, der über klar definierte APIs kommuniziert und in der Regel kleinen, eigenständigen Teams gehört. Ein Versicherungssystem kann beispielsweise Microservices beinhalten, die Geschäftsfunktionen wie Vertrieb oder Marketing oder Subdomains wie Einkauf, Schadenersatz oder Analytik zugeordnet sind. Zu den Vorteilen von Microservices gehören Agilität, flexible Skalierung, einfache Bereitstellung, wiederverwendbarer Code und Ausfallsicherheit. Weitere Informationen finden Sie unter Integration von Microservices mithilfe serverloser Dienste. AWS

Microservices-Architekturen

Ein Ansatz zur Erstellung einer Anwendung mit unabhängigen Komponenten, die jeden Anwendungsprozess als Microservice ausführen. Diese Microservices kommunizieren über eine klar definierte Schnittstelle mithilfe einfacher APIs. Jeder Microservice in dieser Architektur kann aktualisiert, bereitgestellt und skaliert werden, um den Bedarf an bestimmten Funktionen einer Anwendung zu decken. Weitere Informationen finden Sie unter Implementierung von Microservices auf. AWS

Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

Migration in großem Maßstab

Der Prozess, bei dem der Großteil des Anwendungsportfolios in Wellen in die Cloud verlagert wird, wobei in jeder Welle mehr Anwendungen schneller migriert werden. In dieser Phase werden die bewährten Verfahren und Erkenntnisse aus den früheren Phasen zur Implementierung einer Migrationsfabrik von Teams, Tools und Prozessen zur Optimierung der Migration von Workloads durch Automatisierung und agile Bereitstellung verwendet. Dies ist die dritte Phase der AWS - Migrationsstrategie.

Migrationsfabrik

Funktionsübergreifende Teams, die die Migration von Workloads durch automatisierte, agile Ansätze optimieren. Zu den Teams in der Migrationsabteilung gehören in der Regel Betriebsabläufe, Geschäftsanalysten und Eigentümer, Migrationsingenieure, Entwickler und DevOps Experten, die in Sprints arbeiten. Zwischen 20 und 50 Prozent eines Unternehmensanwendungsportfolios bestehen aus sich wiederholenden Mustern, die durch einen Fabrik-Ansatz optimiert werden können. Weitere Informationen finden Sie in Diskussion über Migrationsfabriken und den Leitfaden zur Cloud-Migration-Fabrik in diesem Inhaltssatz.

Migrationsmetadaten

Die Informationen über die Anwendung und den Server, die für den Abschluss der Migration benötigt werden. Für jedes Migrationsmuster ist ein anderer Satz von Migrationsmetadaten erforderlich. Beispiele für Migrationsmetadaten sind das Zielsubnetz, die Sicherheitsgruppe und AWS das Konto.

Migrationsmuster

Eine wiederholbare Migrationsaufgabe, in der die Migrationsstrategie, das Migrationsziel und die verwendete Migrationsanwendung oder der verwendete Migrationsservice detailliert beschrieben werden. Beispiel: Rehost-Migration zu Amazon EC2 mit AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung, Migrationspriorisierung und Wellenplanung). Das MPA-Tool (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

Migration Readiness Assessment (MRA)

Der Prozess, bei dem mithilfe des AWS CAF Erkenntnisse über den Cloud-Bereitschaftsstatus eines Unternehmens gewonnen, Stärken und Schwächen identifiziert und ein Aktionsplan zur Schließung festgestellter Lücken erstellt wird. Weitere Informationen finden Sie im Benutzerhandbuch für Migration Readiness. MRA ist die erste Phase der AWS - Migrationsstrategie.

Migrationsstrategie

Der Ansatz, der verwendet wurde, um einen Workload auf den AWS Cloud zu migrieren. Weitere Informationen finden Sie im Eintrag 7 Rs in diesem Glossar und unter Mobilisieren Sie Ihr Unternehmen, um umfangreiche Migrationen zu beschleunigen.

ML

Siehe maschinelles Lernen.

Modernisierung

Umwandlung einer veralteten (veralteten oder monolithischen) Anwendung und ihrer Infrastruktur in ein agiles, elastisches und hochverfügbares System in der Cloud, um Kosten zu senken, die Effizienz zu steigern und Innovationen zu nutzen. Weitere Informationen finden Sie unter Strategie zur Modernisierung von Anwendungen in der AWS Cloud.

Bewertung der Modernisierungsfähigkeit

Eine Bewertung, anhand derer festgestellt werden kann, ob die Anwendungen einer Organisation für die Modernisierung bereit sind, Vorteile, Risiken und Abhängigkeiten identifiziert und ermittelt wird, wie gut die Organisation den zukünftigen Status dieser Anwendungen unterstützen kann. Das Ergebnis der Bewertung ist eine Vorlage der Zielarchitektur, eine Roadmap, in der die Entwicklungsphasen und Meilensteine des Modernisierungsprozesses detailliert beschrieben werden, sowie ein Aktionsplan zur Behebung festgestellter Lücken. Weitere Informationen finden Sie unter Evaluierung der Modernisierungsbereitschaft von Anwendungen in der AWS Cloud.

Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter Zerlegen von Monolithen in Microservices.

MPA

Siehe Bewertung des Migrationsportfolios.

MQTT

Siehe Message Queuing-Telemetrietransport.

Mehrklassen-Klassifizierung

Ein Prozess, der dabei hilft, Vorhersagen für mehrere Klassen zu generieren (wobei eines von mehr als zwei Ergebnissen vorhergesagt wird). Ein ML-Modell könnte beispielsweise fragen: „Ist dieses Produkt ein Buch, ein Auto oder ein Telefon?“ oder „Welche Kategorie von Produkten ist für diesen Kunden am interessantesten?“

veränderbare Infrastruktur

Ein Modell, das die bestehende Infrastruktur für Produktionsworkloads aktualisiert und modifiziert. Für eine verbesserte Konsistenz, Zuverlässigkeit und Vorhersagbarkeit empfiehlt das AWS Well-Architected Framework die Verwendung einer unveränderlichen Infrastruktur als bewährte Methode.

O

OAC

Weitere Informationen finden Sie unter Origin Access Control.

OAI

Siehe Zugriffsidentität von Origin.

COM

Siehe organisatorisches Change-Management.

Offline-Migration

Eine Migrationsmethode, bei der der Quell-Workload während des Migrationsprozesses heruntergefahren wird. Diese Methode ist mit längeren Ausfallzeiten verbunden und wird in der Regel für kleine, unkritische Workloads verwendet.

OI

Siehe Betriebsintegration.

OLA

Siehe Vereinbarung auf operativer Ebene.

Online-Migration

Eine Migrationsmethode, bei der der Quell-Workload auf das Zielsystem kopiert wird, ohne offline genommen zu werden. Anwendungen, die mit dem Workload verbunden sind, können während der Migration weiterhin funktionieren. Diese Methode beinhaltet keine bis minimale Ausfallzeit und wird in der Regel für kritische Produktionsworkloads verwendet.

OPC-UA

Siehe Open Process Communications — Unified Architecture.

Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

Ein machine-to-machine (M2M) -Kommunikationsprotokoll für die industrielle Automatisierung. OPC-UA bietet einen Interoperabilitätsstandard mit Datenverschlüsselungs-, Authentifizierungs- und Autorisierungsschemata.

Vereinbarung auf Betriebsebene (OLA)

Eine Vereinbarung, in der klargestellt wird, welche funktionalen IT-Gruppen sich gegenseitig versprechen zu liefern, um ein Service Level Agreement (SLA) zu unterstützen.

Überprüfung der Betriebsbereitschaft (ORR)

Eine Checkliste mit Fragen und zugehörigen bewährten Methoden, die Ihnen helfen, Vorfälle und mögliche Ausfälle zu verstehen, zu bewerten, zu verhindern oder deren Umfang zu reduzieren. Weitere Informationen finden Sie unter Operational Readiness Reviews (ORR) im AWS Well-Architected Framework.

Betriebstechnologie (OT)

Hardware- und Softwaresysteme, die mit der physischen Umgebung zusammenarbeiten, um industrielle Abläufe, Ausrüstung und Infrastruktur zu steuern. In der Fertigung ist die Integration

von OT- und Informationstechnologie (IT) -Systemen ein zentraler Schwerpunkt der Industrie 4.0-Transformationen.

Betriebsintegration (OI)

Der Prozess der Modernisierung von Abläufen in der Cloud, der Bereitschaftsplanung, Automatisierung und Integration umfasst. Weitere Informationen finden Sie im Leitfaden zur Betriebsintegration.

Organisationspfad

Ein Pfad, der von erstellt wird und in AWS CloudTrail dem alle Ereignisse für alle AWS-Konten in einer Organisation protokolliert werden. AWS Organizations Diese Spur wird in jedem AWS-Konto , der Teil der Organisation ist, erstellt und verfolgt die Aktivität in jedem Konto. Weitere Informationen finden Sie in der CloudTrail Dokumentation unter Erstellen eines Pfads für eine Organisation.

Organisatorisches Veränderungsmanagement (OCM)

Ein Framework für das Management wichtiger, disruptiver Geschäftstransformationen aus Sicht der Mitarbeiter, der Kultur und der Führung. OCM hilft Organisationen dabei, sich auf neue Systeme und Strategien vorzubereiten und auf diese umzustellen, indem es die Akzeptanz von Veränderungen beschleunigt, Übergangsprobleme angeht und kulturelle und organisatorische Veränderungen vorantreibt. In der AWS Migrationsstrategie wird dieses Framework aufgrund der Geschwindigkeit des Wandels, der bei Projekten zur Cloud-Einführung erforderlich ist, als Mitarbeiterbeschleunigung bezeichnet. Weitere Informationen finden Sie im OCM-Handbuch.

Ursprungszugriffskontrolle (OAC)

In CloudFront, eine erweiterte Option zur Zugriffsbeschränkung, um Ihre Amazon Simple Storage Service (Amazon S3) -Inhalte zu sichern. OAC unterstützt alle S3-Buckets insgesamt AWS-Regionen, serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) sowie dynamische `PUT` und `DELETE` Anfragen an den S3-Bucket.

Ursprungszugriffsidentität (OAI)

In CloudFront, eine Option zur Zugriffsbeschränkung, um Ihre Amazon S3 S3-Inhalte zu sichern. Wenn Sie OAI verwenden, CloudFront erstellt es einen Principal, mit dem sich Amazon S3 authentifizieren kann. Authentifizierte Principals können nur über eine bestimmte Distribution auf Inhalte in einem S3-Bucket zugreifen. CloudFront Siehe auch OAC, das eine detailliertere und verbesserte Zugriffskontrolle bietet.

ODER

Siehe Überprüfung der Betriebsbereitschaft.

NICHT

Siehe Betriebstechnologie.

Ausgehende (egress) VPC

In einer Architektur AWS mit mehreren Konten eine VPC, die Netzwerkverbindungen verarbeitet, die von einer Anwendung aus initiiert werden. Die AWS -Referenzarchitektur für die Sicherheit empfiehlt, Ihr Netzwerkkonto mit eingehenden und ausgehenden VPCs und Inspektions-VPCs einzurichten, um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet zu schützen.

P

Berechtigungsgrenze

Eine IAM-Verwaltungsrichtlinie, die den IAM-Prinzipalen zugeordnet ist, um die maximalen Berechtigungen festzulegen, die der Benutzer oder die Rolle haben kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitys in der IAM-Dokumentation.

persönlich identifizierbare Informationen (PII)

Informationen, die, wenn sie direkt betrachtet oder mit anderen verwandten Daten kombiniert werden, verwendet werden können, um vernünftige Rückschlüsse auf die Identität einer Person zu ziehen. Beispiele für personenbezogene Daten sind Namen, Adressen und Kontaktinformationen.

Personenbezogene Daten

Siehe persönlich identifizierbare Informationen.

Playbook

Eine Reihe vordefinierter Schritte, die die mit Migrationen verbundenen Aufgaben erfassen, z. B. die Bereitstellung zentraler Betriebsfunktionen in der Cloud. Ein Playbook kann die Form von Skripten, automatisierten Runbooks oder einer Zusammenfassung der Prozesse oder Schritte annehmen, die für den Betrieb Ihrer modernisierten Umgebung erforderlich sind.

PLC

Siehe programmierbare Logiksteuerung.

PLM

Siehe Produktlebenszyklusmanagement.

policy

Ein Objekt, das Berechtigungen definieren (siehe identitätsbasierte Richtlinie), Zugriffsbedingungen spezifizieren (siehe ressourcenbasierte Richtlinie) oder die maximalen Berechtigungen für alle Konten in einer Organisation definieren kann AWS Organizations (siehe Dienststeuerungsrichtlinie).

Polyglotte Beharrlichkeit

Unabhängige Auswahl der Datenspeichertechnologie eines Microservices auf der Grundlage von Datenzugriffsmustern und anderen Anforderungen. Wenn Ihre Microservices über dieselbe Datenspeichertechnologie verfügen, kann dies zu Implementierungsproblemen oder zu Leistungseinbußen führen. Microservices lassen sich leichter implementieren und erzielen eine bessere Leistung und Skalierbarkeit, wenn sie den Datenspeicher verwenden, der ihren Anforderungen am besten entspricht. Weitere Informationen finden Sie unter Datenpersistenz in Microservices aktivieren.

Portfoliobewertung

Ein Prozess, bei dem das Anwendungsportfolio ermittelt, analysiert und priorisiert wird, um die Migration zu planen. Weitere Informationen finden Sie in Bewerten der Migrationsbereitschaft.

predicate

Eine Abfragebedingung, die `true` oder zurückgibt`false`, was üblicherweise in einer Klausel vorkommt. `WHERE`

Prädikat Pushdown

Eine Technik zur Optimierung von Datenbankabfragen, bei der die Daten in der Abfrage vor der Übertragung gefiltert werden. Dadurch wird die Datenmenge reduziert, die aus der relationalen Datenbank abgerufen und verarbeitet werden muss, und die Abfrageleistung wird verbessert.

Präventive Kontrolle

Eine Sicherheitskontrolle, die verhindern soll, dass ein Ereignis eintritt. Diese Kontrollen stellen eine erste Verteidigungslinie dar, um unbefugten Zugriff oder unerwünschte Änderungen an Ihrem Netzwerk zu verhindern. Weitere Informationen finden Sie unter Präventive Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Prinzipal

Eine Entität AWS , die Aktionen ausführen und auf Ressourcen zugreifen kann. Bei dieser Entität handelt es sich in der Regel um einen Root-Benutzer für eine AWS-Konto, eine IAM-Rolle oder einen Benutzer. Weitere Informationen finden Sie unter Prinzipal in Rollenbegriffe und -konzepte in der IAM-Dokumentation.

Datenschutz durch Design

Ein Ansatz in der Systemtechnik, der den Datenschutz während des gesamten Engineering-Prozesses berücksichtigt.

Privat gehostete Zonen

Ein Container, der Informationen darüber enthält, wie Amazon Route 53 auf DNS-Abfragen für eine Domain und ihre Subdomains innerhalb einer oder mehrerer VPCs reagieren soll. Weitere Informationen finden Sie unter Arbeiten mit privat gehosteten Zonen in der Route-53-Dokumentation.

proaktive Steuerung

Eine Sicherheitskontrolle, die den Einsatz nicht richtlinienkonformer Ressourcen verhindern soll. Mit diesen Steuerelementen werden Ressourcen gescannt, bevor sie bereitgestellt werden. Wenn die Ressource nicht mit der Steuerung konform ist, wird sie nicht bereitgestellt. Weitere Informationen finden Sie im Referenzhandbuch zu Kontrollen in der AWS Control Tower Dokumentation und unter Proaktive Kontrollen unter Implementierung von Sicherheitskontrollen am AWS.

Produktlebenszyklusmanagement (PLM)

Das Management von Daten und Prozessen für ein Produkt während seines gesamten Lebenszyklus, vom Design, der Entwicklung und Markteinführung über Wachstum und Reife bis hin zur Markteinführung und Markteinführung.

Produktionsumgebung

Siehe Umgebung.

Speicherprogrammierbare Steuerung (SPS)

In der Fertigung ein äußerst zuverlässiger, anpassungsfähiger Computer, der Maschinen überwacht und Fertigungsprozesse automatisiert.

Pseudonymisierung

Der Prozess, bei dem persönliche Identifikatoren in einem Datensatz durch Platzhalterwerte ersetzt werden. Pseudonymisierung kann zum Schutz der Privatsphäre beitragen. Pseudonymisierte Daten gelten weiterhin als personenbezogene Daten.

veröffentlichen/abonnieren (pub/sub)

Ein Muster, das asynchrone Kommunikation zwischen Microservices ermöglicht, um die Skalierbarkeit und Reaktionsfähigkeit zu verbessern. In einem auf Microservices basierenden MES kann ein Microservice beispielsweise Ereignismeldungen in einem Kanal veröffentlichen, den andere Microservices abonnieren können. Das System kann neue Microservices hinzufügen, ohne den Veröffentlichungsservice zu ändern.

Q

Abfrageplan

Eine Reihe von Schritten, wie Anweisungen, die für den Zugriff auf die Daten in einem relationalen SQL-Datenbanksystem verwendet werden.

Abfrageplanregression

Wenn ein Datenbankserviceoptimierer einen weniger optimalen Plan wählt als vor einer bestimmten Änderung der Datenbankumgebung. Dies kann durch Änderungen an Statistiken, Beschränkungen, Umgebungseinstellungen, Abfrageparameter-Bindungen und Aktualisierungen der Datenbank-Engine verursacht werden.

R

RACI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

Ransomware

Eine bösartige Software, die entwickelt wurde, um den Zugriff auf ein Computersystem oder Daten zu blockieren, bis eine Zahlung erfolgt ist.

RASCI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

RCAC

Siehe Zugriffskontrolle für Zeilen und Spalten.

Read Replica

Eine Kopie einer Datenbank, die nur für Lesezwecke verwendet wird. Sie können Abfragen an das Lesereplikat weiterleiten, um die Belastung auf Ihrer Primärdatenbank zu reduzieren.

neu strukturieren

Siehe 7 Rs.

Recovery Point Objective (RPO)

Die maximal zulässige Zeitspanne seit dem letzten Datenwiederherstellungspunkt. Dies bestimmt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Betriebsunterbrechung angesehen wird.

Wiederherstellungszeitziel (RTO)

Die maximal zulässige Verzögerung zwischen der Betriebsunterbrechung und der Wiederherstellung des Dienstes.

Refaktorierung

Siehe 7 Rs.

Region

Eine Sammlung von AWS Ressourcen in einem geografischen Gebiet. Jeder AWS-Region ist isoliert und unabhängig von den anderen, um Fehlertoleranz, Stabilität und Belastbarkeit zu gewährleisten. Weitere Informationen finden Sie unter Geben Sie an, was AWS-Regionen Ihr Konto verwenden kann.

Regression

Eine ML-Technik, die einen numerischen Wert vorhersagt. Zum Beispiel, um das Problem „Zu welchem Preis wird dieses Haus verkauft werden?“ zu lösen Ein ML-Modell könnte ein lineares Regressionsmodell verwenden, um den Verkaufspreis eines Hauses auf der Grundlage bekannter Fakten über das Haus (z. B. die Quadratmeterzahl) vorherzusagen.

rehosten

Siehe 7 Rs.

Veröffentlichung

In einem Bereitstellungsprozess der Akt der Förderung von Änderungen an einer Produktionsumgebung.

umziehen

Siehe 7 Rs.

neue Plattform

Siehe 7 Rs.

Rückkauf

Siehe 7 Rs.

Ausfallsicherheit

Die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich von ihnen zu erholen. Hochverfügbarkeit und Notfallwiederherstellung sind häufig Überlegungen bei der Planung der Ausfallsicherheit in der. AWS Cloud Weitere Informationen finden Sie unter AWS Cloud Resilienz.

Ressourcenbasierte Richtlinie

Eine mit einer Ressource verknüpfte Richtlinie, z. B. ein Amazon-S3-Bucket, ein Endpunkt oder ein Verschlüsselungsschlüssel. Diese Art von Richtlinie legt fest, welchen Prinzipalen der Zugriff gewährt wird, welche Aktionen unterstützt werden und welche anderen Bedingungen erfüllt sein müssen.

RACI-Matrix (verantwortlich, rechenschaftspflichtig, konsultiert, informiert)

Eine Matrix, die die Rollen und Verantwortlichkeiten aller an Migrationsaktivitäten und Cloud-Operationen beteiligten Parteien definiert. Der Matrixname leitet sich von den in der Matrix definierten Zuständigkeitstypen ab: verantwortlich (R), rechenschaftspflichtig (A), konsultiert (C) und informiert (I). Der Unterstützungstyp (S) ist optional. Wenn Sie Unterstützung einbeziehen, wird die Matrix als RASCI-Matrix bezeichnet, und wenn Sie sie ausschließen, wird sie als RACI-Matrix bezeichnet.

Reaktive Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, die Behebung unerwünschter Ereignisse oder Abweichungen von Ihren Sicherheitsstandards voranzutreiben. Weitere Informationen finden Sie unter Reaktive Kontrolle in Implementieren von Sicherheitskontrollen in AWS.

Beibehaltung

Siehe 7 Rs.

zurückziehen

Siehe 7 Rs.

Drehung

Der Vorgang, bei dem ein Geheimnis regelmäßig aktualisiert wird, um es einem Angreifer zu erschweren, auf die Anmeldeinformationen zuzugreifen.

Zugriffskontrolle für Zeilen und Spalten (RCAC)

Die Verwendung einfacher, flexibler SQL-Ausdrücke mit definierten Zugriffsregeln. RCAC besteht aus Zeilenberechtigungen und Spaltenmasken.

RPO

Siehe Recovery Point Objective.

RTO

Siehe Ziel der Wiederherstellungszeit.

Runbook

Eine Reihe manueller oder automatisierter Verfahren, die zur Ausführung einer bestimmten Aufgabe erforderlich sind. Diese sind in der Regel darauf ausgelegt, sich wiederholende Operationen oder Verfahren mit hohen Fehlerquoten zu rationalisieren.

S

SAML 2.0

Ein offener Standard, den viele Identitätsanbieter (IdPs) verwenden. Diese Funktion ermöglicht föderiertes Single Sign-On (SSO), sodass sich Benutzer bei den API-Vorgängen anmelden AWS Management Console oder die AWS API-Operationen aufrufen können, ohne dass Sie einen Benutzer in IAM für alle in Ihrer Organisation erstellen müssen. Weitere Informationen zum SAML-2.0.-basierten Verbund finden Sie unter Über den SAML-2.0-basierten Verbund in der IAM-Dokumentation.

SCADA

Siehe Aufsichtskontrolle und Datenerfassung.

SCP

Siehe Richtlinie zur Dienstkontrolle.

Secret

Interne AWS Secrets Manager, vertrauliche oder eingeschränkte Informationen, wie z. B. ein Passwort oder Benutzeranmeldeinformationen, die Sie in verschlüsselter Form speichern. Es besteht aus dem geheimen Wert und seinen Metadaten. Der geheime Wert kann binär, eine einzelne Zeichenfolge oder mehrere Zeichenketten sein. Weitere Informationen finden Sie unter Was ist in einem Secrets Manager Manager-Geheimnis? in der Secrets Manager Manager-Dokumentation.

Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: präventiv, detektiv, reaktionsschnell und proaktiv.

Härtung der Sicherheit

Der Prozess, bei dem die Angriffsfläche reduziert wird, um sie widerstandsfähiger gegen Angriffe zu machen. Dies kann Aktionen wie das Entfernen von Ressourcen, die nicht mehr benötigt werden, die Implementierung der bewährten Sicherheitsmethode der Gewährung geringster Berechtigungen oder die Deaktivierung unnötiger Feature in Konfigurationsdateien umfassen.

System zur Verwaltung von Sicherheitsinformationen und Ereignissen (security information and event management – SIEM)

Tools und Services, die Systeme für das Sicherheitsinformationsmanagement (SIM) und das Management von Sicherheitsereignissen (SEM) kombinieren. Ein SIEM-System sammelt, überwacht und analysiert Daten von Servern, Netzwerken, Geräten und anderen Quellen, um Bedrohungen und Sicherheitsverletzungen zu erkennen und Warnmeldungen zu generieren.

Automatisierung von Sicherheitsreaktionen

Eine vordefinierte und programmierte Aktion, die darauf ausgelegt ist, automatisch auf ein Sicherheitsereignis zu reagieren oder es zu beheben. Diese Automatisierungen dienen als detektive oder reaktionsschnelle Sicherheitskontrollen, die Sie bei der Implementierung bewährter AWS Sicherheitsmethoden unterstützen. Beispiele für automatisierte Antwortaktionen sind das Ändern einer VPC-Sicherheitsgruppe, das Patchen einer Amazon EC2 EC2-Instance oder das Rotieren von Anmeldeinformationen.

Serverseitige Verschlüsselung

Verschlüsselung von Daten am Zielort durch denjenigen AWS -Service , der sie empfängt.

Service-Kontrollrichtlinie (SCP)

Eine Richtlinie, die eine zentrale Kontrolle über die Berechtigungen für alle Konten in einer Organisation in AWS Organizations ermöglicht. SCPs definieren Integritätsschutz oder legen Grenzwerte für Aktionen fest, die ein Administrator an Benutzer oder Rollen delegieren kann. Sie können SCPs als Zulassungs- oder Ablehnungslisten verwenden, um festzulegen, welche Services oder Aktionen zulässig oder verboten sind. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Richtlinien zur Dienststeuerung.

Service-Endpunkt

Die URL des Einstiegspunkts für einen AWS -Service. Sie können den Endpunkt verwenden, um programmgesteuert eine Verbindung zum Zielservice herzustellen. Weitere Informationen finden Sie unter AWS -Service -Endpunkte in der Allgemeine AWS-Referenz.

Service Level Agreement (SLA)

Eine Vereinbarung, in der klargestellt wird, was ein IT-Team seinen Kunden zu bieten verspricht, z. B. in Bezug auf Verfügbarkeit und Leistung der Services.

Service-Level-Indikator (SLI)

Eine Messung eines Leistungsaspekts eines Dienstes, z. B. seiner Fehlerrate, Verfügbarkeit oder Durchsatz.

Service-Level-Ziel (SLO)

Eine Zielkennzahl, die den Zustand eines Dienstes darstellt, gemessen anhand eines Service-Level-Indikators.

Modell der geteilten Verantwortung

Ein Modell, das die Verantwortung beschreibt, mit der Sie gemeinsam AWS für Cloud-Sicherheit und Compliance verantwortlich sind. AWS ist für die Sicherheit der Cloud verantwortlich, wohingegen Sie für die Sicherheit in der Cloud verantwortlich sind. Weitere Informationen finden Sie unter Modell der geteilten Verantwortung.

SIEM

Siehe Sicherheitsinformations- und Event-Management-System.

Single Point of Failure (SPOF)

Ein Fehler in einer einzelnen, kritischen Komponente einer Anwendung, der das System stören kann.

SLA

Siehe Service Level Agreement.

SLI

Siehe Service-Level-Indikator.

ALSO

Siehe Service-Level-Ziel.

split-and-seed Modell

Ein Muster für die Skalierung und Beschleunigung von Modernisierungsprojekten. Sobald neue Features und Produktversionen definiert werden, teilt sich das Kernteam auf, um neue Produktteams zu bilden. Dies trägt zur Skalierung der Fähigkeiten und Services Ihrer Organisation bei, verbessert die Produktivität der Entwickler und unterstützt schnelle Innovationen. Weitere Informationen finden Sie unter Schrittweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud.

SPOTTEN

Siehe Single Point of Failure.

Sternschema

Eine Datenbank-Organisationsstruktur, die eine große Faktentabelle zum Speichern von Transaktions- oder Messdaten und eine oder mehrere kleinere dimensionale Tabellen zum Speichern von Datenattributen verwendet. Diese Struktur ist für die Verwendung in einem Data Warehouse oder für Business Intelligence-Zwecke konzipiert.

Strangler-Fig-Muster

Ein Ansatz zur Modernisierung monolithischer Systeme, bei dem die Systemfunktionen schrittweise umgeschrieben und ersetzt werden, bis das Legacy-System außer Betrieb genommen werden kann. Dieses Muster verwendet die Analogie einer Feigenrebe, die zu einem etablierten Baum heranwächst und schließlich ihren Wirt überwindet und ersetzt. Das Muster wurde eingeführt von Martin Fowler als Möglichkeit, Risiken beim Umschreiben monolithischer Systeme zu managen. Ein Beispiel für die Anwendung dieses Musters finden Sie

unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

Subnetz

Ein Bereich von IP-Adressen in Ihrer VPC. Ein Subnetz muss sich in einer einzigen Availability Zone befinden.

Aufsichtskontrolle und Datenerfassung (SCADA)

In der Fertigung ein System, das Hardware und Software zur Überwachung von Sachanlagen und Produktionsabläufen verwendet.

Symmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der denselben Schlüssel zum Verschlüsseln und Entschlüsseln der Daten verwendet.

synthetisches Testen

Testen eines Systems auf eine Weise, die Benutzerinteraktionen simuliert, um potenzielle Probleme zu erkennen oder die Leistung zu überwachen. Sie können Amazon CloudWatch Synthetics verwenden, um diese Tests zu erstellen.

T

tags

Schlüssel-Wert-Paare, die als Metadaten für die Organisation Ihrer Ressourcen dienen. AWS Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Weitere Informationen finden Sie unter Markieren Ihrer AWS -Ressourcen.

Zielvariable

Der Wert, den Sie in überwachtem ML vorhersagen möchten. Dies wird auch als Ergebnisvariable bezeichnet. In einer Fertigungsumgebung könnte die Zielvariable beispielsweise ein Produktfehler sein.

Aufgabenliste

Ein Tool, das verwendet wird, um den Fortschritt anhand eines Runbooks zu verfolgen. Eine Aufgabenliste enthält eine Übersicht über das Runbook und eine Liste mit allgemeinen Aufgaben, die erledigt werden müssen. Für jede allgemeine Aufgabe werden der geschätzte Zeitaufwand, der Eigentümer und der Fortschritt angegeben.

Testumgebungen

Siehe Umgebung.

Training

Daten für Ihr ML-Modell bereitstellen, aus denen es lernen kann. Die Trainingsdaten müssen die richtige Antwort enthalten. Der Lernalgorithmus findet Muster in den Trainingsdaten, die die Attribute der Input-Daten dem Ziel (die Antwort, die Sie voraussagen möchten) zuordnen. Es gibt ein ML-Modell aus, das diese Muster erfasst. Sie können dann das ML-Modell verwenden, um Voraussagen für neue Daten zu erhalten, bei denen Sie das Ziel nicht kennen.

Transit-Gateway

Ein Transit-Gateway ist ein Netzwerk-Transit-Hub, mit dem Sie Ihre VPCs und On-Premises-Netzwerke miteinander verbinden können. Weitere Informationen finden Sie in der AWS Transit Gateway Dokumentation unter Was ist ein Transit-Gateway.

Stammbasierter Workflow

Ein Ansatz, bei dem Entwickler Feature lokal in einem Feature-Zweig erstellen und testen und diese Änderungen dann im Hauptzweig zusammenführen. Der Hauptzweig wird dann sequentiell für die Entwicklungs-, Vorproduktions- und Produktionsumgebungen erstellt.

Vertrauenswürdiger Zugriff

Gewährung von Berechtigungen für einen Dienst, den Sie angeben, um Aufgaben in Ihrer Organisation AWS Organizations und in deren Konten in Ihrem Namen auszuführen. Der vertrauenswürdige Service erstellt in jedem Konto eine mit dem Service verknüpfte Rolle, wenn diese Rolle benötigt wird, um Verwaltungsaufgaben für Sie auszuführen. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Verwendung AWS Organizations mit anderen AWS Diensten.

Optimieren

Aspekte Ihres Trainingsprozesses ändern, um die Genauigkeit des ML-Modells zu verbessern. Sie können das ML-Modell z. B. trainieren, indem Sie einen Beschriftungssatz generieren, Beschriftungen hinzufügen und diese Schritte dann mehrmals unter verschiedenen Einstellungen wiederholen, um das Modell zu optimieren.

Zwei-Pizzen-Team

Ein kleines DevOps Team, das Sie mit zwei Pizzen ernähren können. Eine Teamgröße von zwei Pizzen gewährleistet die bestmögliche Gelegenheit zur Zusammenarbeit bei der Softwareentwicklung.

U

Unsicherheit

Ein Konzept, das sich auf ungenaue, unvollständige oder unbekannte Informationen bezieht, die die Zuverlässigkeit von prädiktiven ML-Modellen untergraben können. Es gibt zwei Arten von Unsicherheit: Epistemische Unsicherheit wird durch begrenzte, unvollständige Daten verursacht, wohingegen aleatorische Unsicherheit durch Rauschen und Randomisierung verursacht wird, die in den Daten liegt. Weitere Informationen finden Sie im Leitfaden Quantifizieren der Unsicherheit in Deep-Learning-Systemen.

undifferenzierte Aufgaben

Diese Arbeit wird auch als Schwerstarbeit bezeichnet. Dabei handelt es sich um Arbeiten, die zwar für die Erstellung und den Betrieb einer Anwendung erforderlich sind, aber dem Endbenutzer keinen direkten Mehrwert bieten oder keinen Wettbewerbsvorteil bieten. Beispiele für undifferenzierte Aufgaben sind Beschaffung, Wartung und Kapazitätsplanung.

höhere Umgebungen

Siehe Umgebung.

V

Vacuuming

Ein Vorgang zur Datenbankwartung, bei dem die Datenbank nach inkrementellen Aktualisierungen bereinigt wird, um Speicherplatz zurückzugewinnen und die Leistung zu verbessern.

Versionskontrolle

Prozesse und Tools zur Nachverfolgung von Änderungen, z. B. Änderungen am Quellcode in einem Repository.

VPC-Peering

Eine Verbindung zwischen zwei VPCs, mit der Sie den Datenverkehr mithilfe von privaten IP-Adressen weiterleiten können. Weitere Informationen finden Sie unter Was ist VPC-Peering? in der Amazon-VPC-Dokumentation.

Schwachstelle

Ein Software- oder Hardwarefehler, der die Sicherheit des Systems gefährdet.

W

Warmer Cache

Ein Puffer-Cache, der aktuelle, relevante Daten enthält, auf die häufig zugegriffen wird. Die Datenbank-Instance kann aus dem Puffer-Cache lesen, was schneller ist als das Lesen aus dem Hauptspeicher oder von der Festplatte.

warme Daten

Daten, auf die selten zugegriffen wird. Bei der Abfrage dieser Art von Daten sind mäßig langsame Abfragen in der Regel akzeptabel.

Fensterfunktion

Eine SQL-Funktion, die eine Berechnung für eine Gruppe von Zeilen durchführt, die sich in irgendeiner Weise auf den aktuellen Datensatz beziehen. Fensterfunktionen sind nützlich für die Verarbeitung von Aufgaben wie die Berechnung eines gleitenden Durchschnitts oder für den Zugriff auf den Wert von Zeilen auf der Grundlage der relativen Position der aktuellen Zeile.

Workload

Ein Workload ist eine Sammlung von Ressourcen und Code, die einen Unternehmenswert bietet, wie z. B. eine kundenorientierte Anwendung oder ein Backend-Prozess.

Workstream

Funktionsgruppen in einem Migrationsprojekt, die für eine bestimmte Reihe von Aufgaben verantwortlich sind. Jeder Workstream ist unabhängig, unterstützt aber die anderen Workstreams im Projekt. Der Portfolio-Workstream ist beispielsweise für die Priorisierung von Anwendungen, die Wellenplanung und die Erfassung von Migrationsmetadaten verantwortlich. Der Portfolio-Workstream liefert diese Komponenten an den Migrations-Workstream, der dann die Server und Anwendungen migriert.

WURM

Sehen Sie einmal schreiben, viele lesen.

WQF

Weitere Informationen finden Sie unter AWS Workload Qualification Framework.

einmal schreiben, viele lesen (WORM)

Ein Speichermodell, das Daten ein einziges Mal schreibt und verhindert, dass die Daten gelöscht oder geändert werden. Autorisierte Benutzer können die Daten so oft wie nötig lesen, aber sie können sie nicht ändern. Diese Datenspeicherinfrastruktur gilt als unveränderlich.

Z

Zero-Day-Exploit

Ein Angriff, in der Regel Malware, der eine Zero-Day-Sicherheitslücke ausnutzt.

Zero-Day-Sicherheitslücke

Ein unfehlbarer Fehler oder eine Sicherheitslücke in einem Produktionssystem. Bedrohungsakteure können diese Art von Sicherheitslücke nutzen, um das System anzugreifen. Entwickler werden aufgrund des Angriffs häufig auf die Sicherheitsanfälligkeit aufmerksam.

Zombie-Anwendung

Eine Anwendung, deren durchschnittliche CPU- und Arbeitsspeichernutzung unter 5 Prozent liegt. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen.

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.