



Framework für Resilienzanalyse

AWS Präskriptive Leitlinien



AWS Präskriptive Leitlinien: Framework für Resilienzanalyse

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

| | |
|---|----|
| Einführung | 1 |
| Übersicht über das -Framework | 3 |
| Den Workload verstehen | 7 |
| Anwendung des Frameworks | 9 |
| Abschwächung potenzieller Ausfälle | 12 |
| Grundlegendes zu Kompromissen und Risiken | 12 |
| Beobachtbarkeit des Fehlermodus | 14 |
| Allgemeine Strategien zur Abschwächung | 15 |
| Kontinuierliche Verbesserung | 22 |
| Fazit und Ressourcen | 23 |
| Dokumentverlauf | 24 |
| Glossar | 25 |
| # | 25 |
| A | 26 |
| B | 29 |
| C | 31 |
| D | 34 |
| E | 39 |
| F | 41 |
| G | 42 |
| H | 43 |
| I | 44 |
| L | 47 |
| M | 48 |
| O | 52 |
| P | 55 |
| Q | 58 |
| R | 58 |
| S | 61 |
| T | 65 |
| U | 67 |
| V | 67 |
| W | 68 |
| Z | 69 |

..... lxx

Rahmen für die Resilienzanalyse

John Formento, Bruno Emer, Steven Hooper, Jason Barto und Michael Haken, Amazon Web Services (AWS)

September 2023([Historie dokumentieren](#))

Konsistente, wiederholbare Standards und Prozesse sind ein wichtiger Bestandteil der kontinuierlichen Verbesserung. Dies gilt auch für die Widerstandsfähigkeit verteilter Systeme. Der Zweck dieser Leitlinien besteht darin, ein Framework für die Resilienzanalyse einzuführen, das eine konsistente Methode zur Analyse von Ausfallarten und deren möglichen Auswirkungen auf Ihre Workloads bietet. Wenn Sie dieses Framework während des gesamten Lebenszyklus Ihres Workloads, vom Entwurf bis zum Betrieb, verwenden, können Sie die Widerstandsfähigkeit Ihrer Workloads gegenüber einem breiteren Spektrum potenzieller Ausfallarten kontinuierlich auf konsistente und wiederholbare Weise verbessern. Auf diese Weise können Sie sicherstellen, dass Sie Ihre Resilienzziele erreichen und die gewünschten Resilienzeigenschaften Ihrer Workloads beibehalten.

Dieses Framework wurde auf der Grundlage der Erfahrungen der Außendienstteams für AWS-Lösungsarchitektur bei der Arbeit mit Kunden aus allen Branchen entwickelt. Es richtet sich an Entwickler, die viele Berufsbezeichnungen haben können, darunter Produktmanager, Softwareentwickler, Systemingenieure, Betriebsteams und Architekten. Dies sind die Personen, die am meisten über das System, die Dienstleistung oder das Produkt wissen, das analysiert wird. Die Nutzung des Frameworks in kontinuierlichen Übungen kann Ihnen helfen, schrittweise Fortschritte zu erzielen und Ihre langfristigen Resilienzziele zu erreichen.

Der Schwerpunkt des Frameworks liegt auf der Identifizierung potenzieller Fehlerquellen und der präventiven und korrektiven Maßnahmen, mit denen Sie deren Auswirkungen abmildern können. Selbst wenn die Fehler in Komponenten auftreten, die nicht direkt unter Ihrer Kontrolle stehen, wie z. B. erhöhte Fehlerraten in einer Abhängigkeit, müssen Sie bedenken, wie sich diese Ausfälle auf Ihre Arbeitslast auswirken könnten und wie Sie diese Arbeitslast so gestalten können, dass sie auf diese Fehler reagiert. Letztlich sollten Sie sich auf Folgendes konzentrieren: Fehler, auf die Sie reagieren können, indem Sie eine Abhilfemaßnahme verwenden, auf die Sie Einfluss haben.

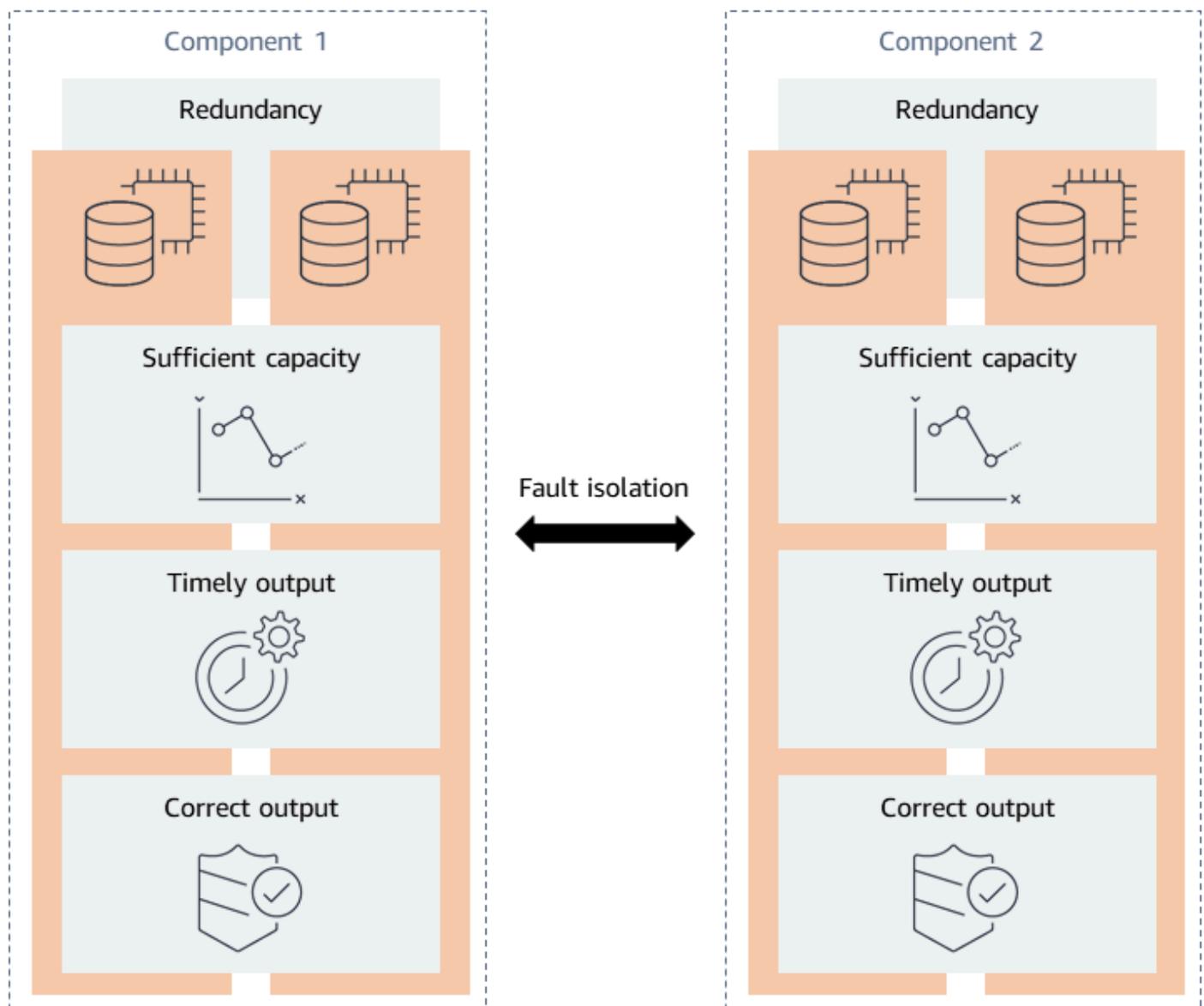
In diesem Leitfaden wird das Framework beschrieben. Anschließend wird erläutert, wie ein Workload identifiziert und dokumentiert wird, wie das Framework auf diesen Workload angewendet wird und wie Strategien zur Risikominderung für mögliche Fehler, die Sie finden, bewertet werden können.

Inhalt

- [Überblick über das Framework](#)
- [Den Arbeitsaufwand verstehen](#)
- [Anwendung des Frameworks](#)
- [Minderung potenzieller Ausfälle](#)
- [Fazit und Ressourcen](#)

Übersicht über das -Framework

Der Rahmen für die Resilienzanalyse wurde entwickelt, indem die gewünschten Resilienzeigenschaften einer Arbeitslast ermittelt wurden. Bei den gewünschten Eigenschaften handelt es sich um die Dinge, die für das System gelten sollen. Resilienz wird in der Regel an der Verfügbarkeit gemessen. Daher sind fünf Eigenschaften die Merkmale eines hochverfügbaren verteilten Systems: Redundanz, ausreichende Kapazität, rechtzeitige Ausgabe, korrekte Ausgabe und Fehlerisolierung. Diese Eigenschaften sind in der folgenden Abbildung dargestellt.



- **Redundanz— Fehlertoleranz** wird durch Redundanz erreicht, die Single Points of Failure (SPOFs) eliminiert. Die Redundanz kann sich von Ersatzkomponenten in Ihrem Workload bis hin zu vollständigen Replikaten Ihres gesamten Anwendungsstapels erstrecken. Wenn Sie Redundanz für Ihre Anwendungen in Betracht ziehen, ist es wichtig, den Grad der Redundanz zu berücksichtigen, den die Infrastruktur, die Datenspeicher und die Abhängigkeiten bieten, die Sie verwenden. Amazon DynamoDB und Amazon Simple Storage Service (Amazon S3) bieten beispielsweise Redundanz, indem sie Daten über mehrere Availability Zones in einer Region replizieren, und AWS Lambda führt Ihre Funktionen über mehrere Worker-Knoten in mehreren Availability Zones aus. Berücksichtigen Sie bei jedem Service, den Sie verwenden, was der Service bietet und wofür Sie entwerfen müssen.
- **Ausreichende Kapazität—** Ihr Workload benötigt ausreichend Ressourcen, um wie vorgesehen zu funktionieren. Zu den Ressourcen gehören Arbeitsspeicher, CPU-Zyklen, Threads, Speicher, Durchsatz, Servicekontingenten und viele andere.
- **Rechtzeitige Ausgabe—** Wenn Kunden Ihren Workload nutzen, erwarten sie, dass er seine vorgesehene Funktion innerhalb eines angemessenen Zeitraums erfüllt. Sofern der Service kein Service Level Agreement (SLA) für die Latenz vorsieht, basieren ihre Erwartungen in der Regel auf empirischen Erkenntnissen, d. h. auf ihren eigenen Erfahrungen. Das durchschnittliche Kundenerlebnis wird normalerweise als die mittlere Latenz (P50) in Ihrem System angesehen. Wenn Ihre Arbeitslast länger als erwartet dauert, kann sich diese Latenz auf das Kundenerlebnis auswirken.
- **Richtige Ausgabe—** Die korrekte Ausgabe der Software Ihres Workloads ist erforderlich, damit dieser die beabsichtigte Funktionalität bereitstellen kann. Ein falsches oder unvollständiges Ergebnis kann schlimmer sein als gar keine Antwort.
- **Isolierung von Fehlern—** Durch die Fehlerisolierung wird der Einflussbereich auf den vorgesehenen Fehlercontainer beschränkt, wenn ein Fehler auftritt. Sie stellt sicher, dass bestimmte Komponenten Ihres Workloads gleichzeitig ausfallen, und verhindert gleichzeitig, dass ein Ausfall auf andere unbeabsichtigte Komponenten übergreift. Es trägt auch dazu bei, den Umfang der Auswirkungen Ihres Workloads auf die Kunden zu begrenzen. Die Fehlerisolierung unterscheidet sich etwas von den vorherigen vier Eigenschaften, da sie akzeptiert, dass ein Fehler bereits aufgetreten ist, dieser aber eingedämmt werden sollte. Sie können eine Fehlerisolierung in Ihrer Infrastruktur, Ihren Abhängigkeiten und Softwarefunktionen einrichten.

Wenn eine gewünschte Eigenschaft verletzt wird, kann dies dazu führen, dass ein Workload nicht verfügbar ist oder als nicht verfügbar wahrgenommen wird. Basierend auf diesen gewünschten Resilienzeigenschaften und unserer Erfahrung in der Zusammenarbeit mit vielen AWS für unsere

Kunden haben wir fünf gängige Fehlerkategorien identifiziert: einzelne Fehlerquellen, übermäßige Auslastung, übermäßige Latenz, Fehlkonfigurationen und Bugs sowie Shared Fate, was wir mit SEEMS abkürzen. Diese bieten eine einheitliche Methode zur Kategorisierung potenzieller Fehlerarten und werden in der folgenden Tabelle beschrieben.

| Kategorie des Fehlers | Verstößt | Definition |
|----------------------------------|------------------------|--|
| Single Points of Failure (SPOFs) | Redundanz | Ein Ausfall einer einzelnen Komponente unterbricht das System aufgrund mangelnder Redundanz der Komponente. |
| Übermäßige Belastung | Ausreichende Kapazität | Ein übermäßiger Verbrauch einer Ressource durch übermäßigen Bedarf oder übermäßigen Verkehr verhindert, dass die Ressource ihre erwartete Funktion erfüllt. Dies kann das Erreichen von Grenzwerten und Kontingenten beinhalten, was zu einer Drosselung und Ablehnung von Anfragen führen kann. |
| Übermäßige Latenz | Rechtzeitige Ausgabe | Die Latenz bei der Systemverarbeitung oder beim Netzwerkverkehr überschreitet die erwartete Zeit, die erwarteten Service Level Objectives (SLOs) oder Service Level Agreements (SLAs). |
| Fehlkonfiguration und Bugs | Korrekte Ausgabe | Softwarefehler oder eine Fehlkonfiguration des Systems |

führen zu einer falschen Ausgabe.

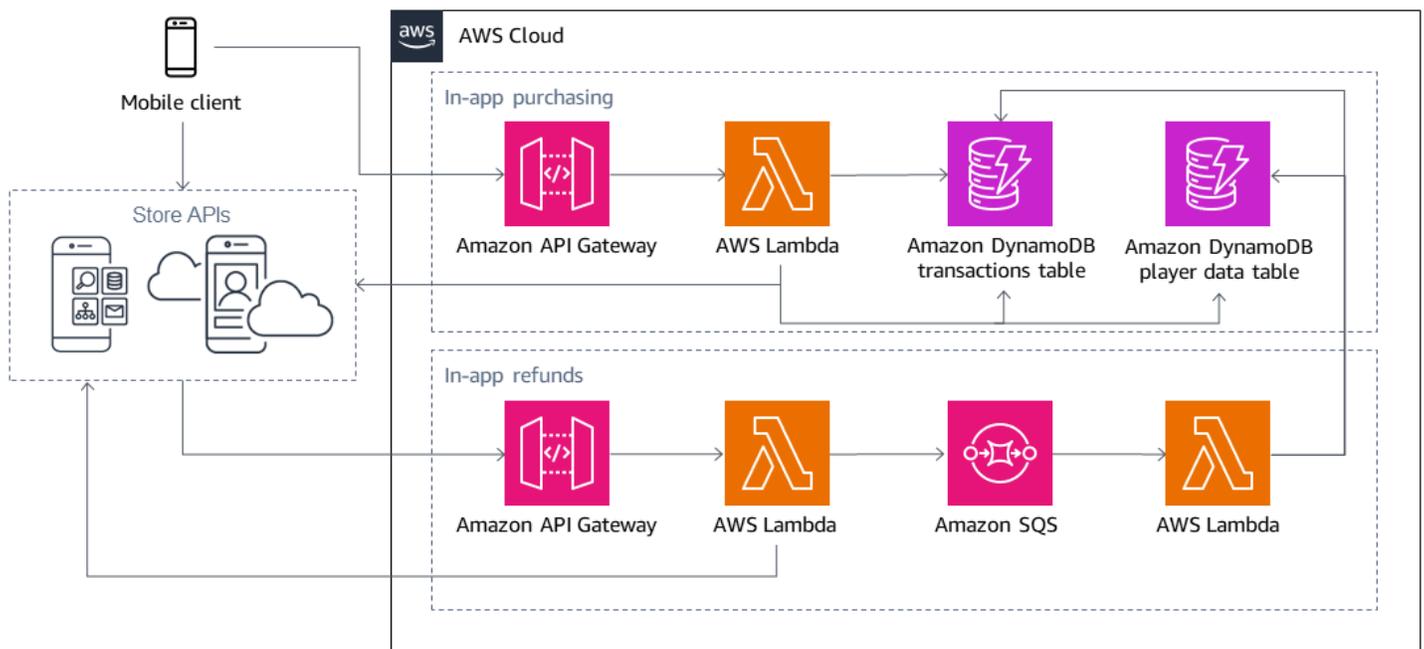
Geteiltes Schicksal

Isolierung von Fehlern

Ein Fehler, der durch eine der vorherigen Fehlerkategorien verursacht wurde, überschreitet die Grenzen der vorgesehenen Fehlerisolierung und überträgt sich auf andere Teile des Systems oder auf andere Kunden.

Den Workload verstehen

Um das Framework anwenden zu können, sollten Sie zunächst die Arbeitslast verstehen, die Sie analysieren möchten. Ein Systemarchitekturdiagramm bietet einen Ausgangspunkt für die Dokumentation der wichtigsten Details des Systems. Der Versuch, eine gesamte Arbeitslast zu analysieren, kann jedoch komplex sein, da viele Systeme aus zahlreichen Komponenten und Interaktionen bestehen. Stattdessen empfehlen wir Ihnen, sich auf Folgendes zu konzentrieren [Geschichten von Benutzern](#), welche sind informelle, allgemeine Erläuterungen der Softwarefunktionen, die aus der Sicht des Endbenutzers verfasst wurden. Ihr Zweck besteht darin, zu verdeutlichen, wie eine Softwarefunktion dem Kunden einen Mehrwert bietet. Anschließend können Sie diese Anwenderberichte mit Architektur- und Datenflussdiagrammen modellieren, um die Bewertung der technischen Komponenten zu erleichtern, die die beschriebene Geschäftsfunktionalität bereitstellen. Eine In-App-Kauflösung für Handyspiele könnte beispielsweise zwei User Stories enthalten: „In-App-Credits kaufen“ und „In-App-Rückerstattungen erhalten“, wie in der folgenden Abbildung dargestellt. (Diese Beispielarchitektur verdeutlicht, wie Sie ein System in User Storys zerlegen können. Sie ist nicht dazu gedacht, eine äußerst robuste Anwendung darzustellen.)



Jede User Story besteht aus vier gemeinsamen Komponenten: Code und Konfiguration, Infrastruktur, Datenspeicher und externe Abhängigkeiten. Ihre Diagramme sollten all diese Komponenten enthalten und die Interaktionen zwischen den Komponenten widerspiegeln. Wenn beispielsweise Ihr Amazon API Gateway-Endpoint überlastet ist, sollten Sie bedenken, wie sich diese Last kaskadierend auf

andere Komponenten im System überträgt, z. B. auf Ihre AWS Lambda-Funktionen oder Amazon DynamoDB-Tabellen. Wenn Sie diese Interaktionen verfolgen, können Sie besser verstehen, wie sich der Fehlermodus auf die User Story auswirken kann. Sie können diesen Fluss visuell mit einem Datenflussdiagramm oder mithilfe einfacher Flusspfeile in einem Architekturdiagramm erfassen, wie in der vorherigen Abbildung. Erwägen Sie, für jede Komponente Details zu erfassen, z. B. die Art der übertragenen Informationen, die empfangenen Informationen, ob die Kommunikation synchron oder asynchron ist und welche Fehlergrenzen überschritten werden. In diesem Beispiel werden die DynamoDB-Tabellen in beiden User Storys gemeinsam genutzt, wie Sie anhand der Pfeile sehen können, die angeben, dass die Lambda-Komponente in der In-App-Rückerstattungsstory auf die DynamoDB-Tabellen in der In-App-Kaufstory zugreift. Das bedeutet, dass ein Fehler, der durch die Benutzerstory zum In-App-Kauf verursacht wird, aufgrund eines gemeinsamen Schicksals zur User Story mit In-App-Rückerstattungen führen kann.

Darüber hinaus ist es wichtig, die Basiskonfiguration für jede Komponente zu verstehen. Die Basiskonfiguration identifiziert Einschränkungen wie die durchschnittliche und maximale Anzahl von Transaktionen pro Sekunde, die maximale Größe einer Nutzlast, ein Client-Timeout und standardmäßige oder aktuelle Dienstkontingente für die Ressource. Wenn Sie einen neuen Entwurf modellieren, empfehlen wir Ihnen, die funktionalen Anforderungen für den Entwurf zu dokumentieren und die Grenzen zu berücksichtigen. Dies hilft Ihnen zu verstehen, wie sich Fehlermodi in der Komponente manifestieren können.

Schließlich sollten Sie User Stories auf der Grundlage des geschäftlichen Nutzens, den sie bieten, priorisieren. Diese Priorisierung hilft Ihnen, sich zunächst auf die wichtigsten Funktionen Ihres Workloads zu konzentrieren. Anschließend können Sie Ihre Analyse auf die Workload-Komponenten konzentrieren, die Teil des kritischen Pfads für diese Funktionalität sind, und so schneller den Nutzen aus der Nutzung des Frameworks ziehen. Während Sie den Prozess durchlaufen, können Sie weitere User Stories mit unterschiedlichen Prioritäten untersuchen.

Anwendung des Frameworks

Die beste Methode, das Framework für die Resilienzanalyse anzuwenden, besteht darin, mit einem Standardsatz von Fragen zu beginnen, die nach Fehlerkategorien geordnet sind und die Sie zu jeder Komponente der User Story stellen sollten, die Sie analysieren. Wenn einige Fragen nicht für jede Komponente Ihres Workloads gelten, verwenden Sie die Fragen, die am besten geeignet sind.

Sie können das Nachdenken über Fehlerarten aus zwei Perspektiven betrachten:

- Wie wirkt sich der Ausfall auf die Fähigkeit der Komponente aus, die User Story zu unterstützen?
- Wie wirkt sich der Ausfall auf die Interaktionen der Komponente mit den anderen Komponenten aus?

Wenn Sie beispielsweise Datenspeicher und übermäßige Last in Betracht ziehen, denken Sie möglicherweise an Fehlermodi, bei denen die Datenbank überlastet ist und bei Abfragen ein Timeout auftritt. Sie könnten auch darüber nachdenken, wie Ihr Datenbankclient die Datenbank mit Wiederholungsversuchen überfordern oder Datenbankverbindungen nicht schließen kann, wodurch der Verbindungspool erschöpft ist. Ein anderes Beispiel ist ein Authentifizierungsprozess, der mehrere Schritte umfassen kann. Sie müssen sich überlegen, wie sich der Ausfall einer Multi-Faktor-Authentifizierungsanwendung (MFA) oder eines externen Identitätsanbieters (IdP) auf eine User Story in diesem Authentifizierungssystem auswirken könnte.

Bei der Beantwortung der folgenden Fragen sollten Sie die Ursache des Fehlers berücksichtigen. Wurde die Überlastung beispielsweise durch einen Kundenansturm oder durch einen menschlichen Bediener verursacht, der während einer Wartungsmaßnahme zu viele Knoten außer Betrieb genommen hat? Möglicherweise können Sie in jeder Frage mehrere Fehlerquellen identifizieren, was unterschiedliche Abhilfemaßnahmen erfordern könnte. Halten Sie beim Stellen der Fragen fest, welche potenziellen Fehlerarten Sie entdeckt haben, für welche Komponente (n) sie gelten und welche Fehlerursachen es gibt.

Einzelne Fehlerquellen

- Ist die Komponente auf Redundanz ausgelegt?
- Was passiert, wenn die Komponente ausfällt?
- Kann Ihre Anwendung den teilweisen oder vollständigen Verlust einer einzelnen Availability Zone tolerieren?

Übermäßige Latenz

- Was passiert, wenn bei dieser Komponente eine erhöhte Latenz auftritt oder eine Komponente, mit der sie interagiert, eine erhöhte Latenz aufweist (oder Netzwerkunterbrechungen wie TCP-Resets)?
- Haben Sie Timeouts mit einer Wiederholungsstrategie entsprechend konfiguriert?
- Scheitern Sie schnell oder langsam? Gibt es kaskadierende Effekte, z. B. das unbeabsichtigte Weiterleiten des gesamten Datenverkehrs an eine beeinträchtigte Ressource, weil diese schnell ausfällt?
- Was sind die teuersten Anfragen an diese Komponente?

Übermäßige Belastung

- Was kann diese Komponente überfordern? Wie kann diese Komponente andere Komponenten überfordern?
- Wie können Sie verhindern, dass Ressourcen für Arbeiten verschwendet werden, die niemals erfolgreich sein werden?
- Haben Sie einen Schutzschalter, der für die Komponente konfiguriert ist?
- Kann etwas zu einem unüberwindbaren Rückstau führen?
- Wo kann diese Komponente bimodales Verhalten erfahren?
- Welche Grenzwerte oder Servicekontingente können überschritten werden (einschließlich Speicherkapazität)?
- Wie skaliert die Komponente unter Last?

Fehlkonfiguration und Bugs

- Wie verhindert man, dass Fehlkonfigurationen und Bugs in der Produktion eingesetzt werden?
- Können Sie eine fehlerhafte Bereitstellung automatisch rückgängig machen oder den Datenverkehr von dem Fehlercontainer wegverlagern, in dem das Update oder die Änderung bereitgestellt wurde?
- Welche Schutzplanken haben Sie eingerichtet, um Bedienungsfehler zu vermeiden?
- Welche Elemente (wie Anmeldeinformationen oder Zertifikate) können ablaufen?

Gemeinsames Schicksal

- Was sind deine Fehler, Isolationsgrenzen?
- Sind die an den Bereitstellungseinheiten vorgenommenen Änderungen mindestens so gering wie beabsichtigt Grenzen der Fehlerisolierung aber idealerweise kleiner, wie z. B. eine One-Box-Umgebung (eine einzelne Instanz innerhalb der Grenze zur Fehlerisolierung)?
- Wird diese Komponente von User Stories oder anderen Workloads gemeinsam genutzt?
- Welche anderen Komponenten sind eng mit dieser Komponente verknüpft?
- Was passiert, wenn bei dieser Komponente oder ihren Abhängigkeiten ein teilweiser oder grauer Fehler auftritt?

Nachdem Sie diese Fragen gestellt haben, können Sie SEEMS auch verwenden, um weitere Fragen zu entwickeln, die für Ihren Workload und die einzelnen Komponenten spezifisch sind. SEEMS eignet sich am besten als strukturierte Methode, um über Ausfallarten nachzudenken, und als Inspirationsquelle bei der Durchführung einer Resilienzanalyse. Es ist keine starre Taxonomie. Vergeuden Sie keine Zeit damit, sich Gedanken darüber zu machen, in welche Kategorie ein bestimmter Fehlermodus passt — das ist nicht wichtig. Was ist wichtig ist, dass du an den Fehler gedacht und ihn aufgeschrieben hast. Es gibt keine falschen Antworten; kreativ zu sein und über den Tellerrand hinauszuschauen ist von Vorteil. Gehen Sie außerdem nicht davon aus, dass ein Fehlermodus bereits behoben ist, sondern beziehen Sie alle möglichen Fehlerarten mit ein, die Ihnen einfallen.

Es ist unwahrscheinlich, dass Sie in Ihrer ersten Übung alle möglichen Ausfallarten antizipieren werden. Mehrere Iterationen des Frameworks helfen Ihnen dabei, ein vollständigeres Modell zu erstellen, sodass Sie nicht versuchen müssen, alles im ersten Durchgang zu lösen. Sie können die Analyse in regelmäßigen, wöchentlichen oder zweiwöchentlichen Abständen ausführen. Konzentrieren Sie sich in jeder Sitzung auf einen bestimmten Fehlermodus oder eine bestimmte Komponente. Dies kann dazu beitragen, stetige, schrittweise Fortschritte bei der Verbesserung der Belastbarkeit Ihrer Arbeitslast zu erzielen. Nachdem Sie eine Liste potenzieller Fehlerquellen für eine User Story zusammengestellt haben, können Sie entscheiden, was Sie dagegen tun möchten.

Abschwächung potenzieller Ausfälle

Jetzt, da Sie potenzielle Fehler für Komponenten in einer Benutzerhistorie haben, können Sie sich auf Abhilfemaßnahmen konzentrieren. Überprüfen Sie zunächst die potenziellen Kompromisse in Bezug auf die potenziellen Auswirkungen und die Wahrscheinlichkeit jedes Fehlers, den Sie entdeckt haben. Bestimmen Sie dann die erforderliche Beobachtbarkeitsstufe und wählen Sie eine Abhilfestrategie aus. Die Kompromisse sollten den Aufwand beinhalten, das richtige Maß an Beobachtbarkeit und Abhilfestrategie zu instrumentieren. Bestimmen Sie abschließend den richtigen Takt für die regelmäßige Überprüfung der Ausfallsicherheit.

Sections

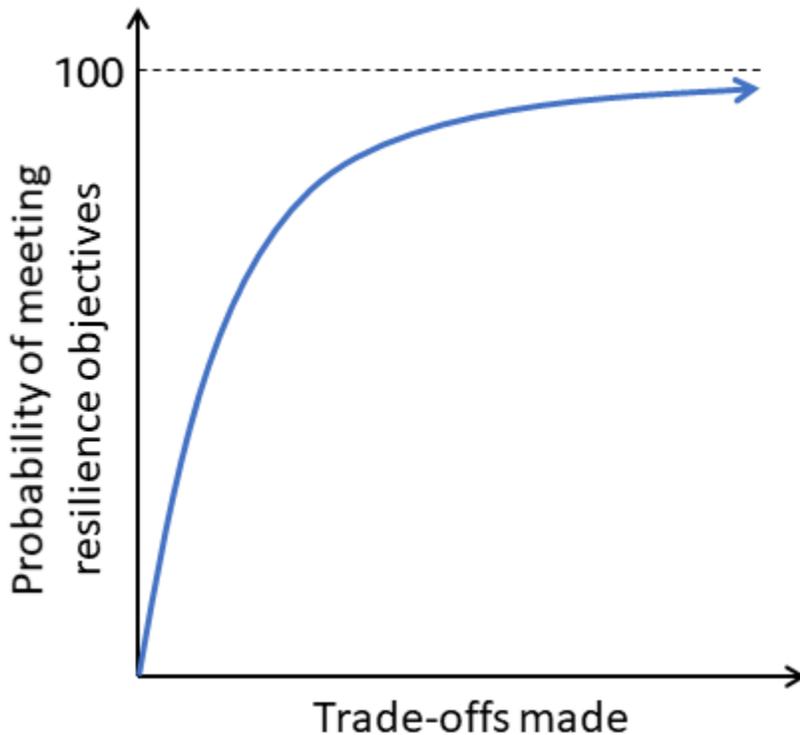
- [Verstehen von Kompromissen und Risiken](#)
- [Beobachtbarkeit des Fehlermodus](#)
- [Allgemeine Strategien zur Abschwächung](#)
- [Kontinuierliche Verbesserung](#)

Grundlegendes zu Kompromissen und Risiken

Ausfallsichere Architekturen sollten eine Handvoll bewährter, einfacher und zuverlässiger Mechanismen verwenden, um auf Ausfälle zu reagieren. Um ein Höchstmaß an Ausfallsicherheit zu erreichen, sollten Workloads so viele Fehlermodi wie möglich automatisch erkennen und wiederherstellen. Dies erfordert umfangreiche Investitionen in die Durchführung von Resilienzanalysen. Das bedeutet, dass eine höhere Resilienz bedeutet, dass Kompromisse eingegangen werden müssen. Wenn Sie jedoch weiterhin Kompromisse eingehen, erreichen Sie einen Punkt, an dem die Rückgaben im Verhältnis zu Ihren Resilienzzielen sinken. Hier sind die typischen Kompromisse:

- Kosten – Redundante Komponenten, verbesserte Beobachtbarkeit, zusätzliche Tools oder eine erhöhte Ressourcennutzung führen zu höheren Kosten.
- Systemkomplexität – Das Erkennen und Reagieren auf Fehlermodi, einschließlich der Abschwächungslösungen, und möglicherweise nicht die Verwendung verwalteter Services, führt zu einer erhöhten Systemkomplexität.
- Entwicklungsaufwand – Für die Entwicklung von Lösungen zur Erkennung von Fehlermodi und zur Reaktion darauf sind zusätzliche Entwicklerstunden erforderlich.

- Betriebsaufwand – Die Überwachung und der Betrieb eines Systems, das mehr Fehlermodi verarbeitet, kann zu Betriebsaufwand führen, insbesondere wenn Sie verwaltete Services nicht verwenden können, um bestimmte Fehlermodi zu minimieren.
- Latenz und Konsistenz – Der Aufbau verteilter Systeme, die Verfügbarkeit bevorzugen, erfordert Kompromisse in Bezug auf Konsistenz und Latenz, wie im [PACELC-Theorem](#) beschrieben.



Berücksichtigen Sie bei der Berücksichtigung der Abhilfemaßnahmen für die identifizierten Fehlermodi in der User History die Kompromisse, die Sie treffen müssen. Wie bei der Sicherheit ist Resilienz ein Optimierungsproblem. Sie müssen eine Entscheidung darüber treffen, ob Sie die Risiken vermeiden, mindern, übertragen oder akzeptieren sollen, die sich aus dem identifizierten Fehler ergeben. Es kann einige Fehlermodi geben, die Sie vermeiden können, einen Satz, den Sie akzeptieren, und einige, die Sie übertragen können. Sie können sich dafür entscheiden, viele der von Ihnen identifizierten Fehlermodi zu minimieren. Um zu bestimmen, welchen Ansatz Sie wählen sollten, führen Sie eine Bewertung durch, indem Sie zwei Fragen stellen: Wie hoch ist die Wahrscheinlichkeit, dass der Fehler auftritt? Welche Auswirkungen hat dies auf den Workload?

Bei Likeli microSD handelt es sich um die Art und Weise, wie ein Ereignis eintritt. Wenn die Benutzererfahrung beispielsweise eine Komponente hat, die auf einer einzelnen Amazon Elastic Compute Cloud (Amazon EC2)-Instance funktioniert, kann die Komponente irgendwann während

des Betriebs des Systems unterbrochen werden, möglicherweise aufgrund von Patching-Verfahren oder Betriebssystemfehlern. Alternativ hat eine Datenbank, die von Amazon Relational Database Service (Amazon RDS) verwaltet wird und Daten zwischen ihren primären und sekundären Instances synchronisiert, eine geringe Wahrscheinlichkeit, vollständig nicht verfügbar zu sein.

Auswirkung ist eine Schätzung des Gesundheitszustands, den ein Ereignis verursachen kann. Sie sollte sowohl aus finanzieller als auch aus Reputationssicht bewertet werden und bezieht sich auf den Wert der Benutzererfahrungen, auf die sie sich auswirkt. Beispielsweise könnte eine überlastete Datenbank erhebliche Auswirkungen auf die Fähigkeit eines E-Commerce-Systems haben, neue Bestellungen anzunehmen. Der Verlust einer einzelnen Instance aus einer Flotte von 20 Instances hinter einem Load Balancer hätte jedoch wahrscheinlich sehr geringe Auswirkungen.

Sie können die Antworten auf diese Fragen mit den Kosten der Kompromisse vergleichen, die Sie ergreifen müssen, um das Risiko zu minimieren. Wenn Sie diese Informationen im Hinblick auf Ihren Risikoschwellenwert und Ihre Resilienzziele berücksichtigen, trifft sie Ihre Entscheidung darüber, welche Fehlermodi Sie aktiv minimieren möchten.

Beobachtbarkeit des Fehlermodus

Um einen Fehlermodus zu minimieren, müssen Sie zunächst erkennen, dass er sich derzeit auf Ihren Workload auswirkt oder bald beeinträchtigt wird. Eine Abhilfemaßnahme ist nur wirksam, wenn ein Signal dafür vorliegt, dass eine Aktion ausgeführt werden muss. Dies bedeutet, dass ein Teil der Erstellung einer Abhilfe mindestens die Überprüfung umfasst, ob Sie die Beobachtbarkeit haben oder aufbauen, die erforderlich ist, um die Auswirkungen des Fehlers zu erkennen.

Sie sollten die beobachtbaren Merkmale des Fehlermodus in zwei Dimensionen berücksichtigen:

- Was sind die wichtigsten Indikatoren, die Sie darüber informieren, dass sich das System einer Bedingung nähert, in der bald Auswirkungen auftreten könnten?
- Was sind die Verzögerungsindikatoren, die die Auswirkungen des Fehlermodus so schnell wie möglich anzeigen können, nachdem er eingetreten ist?

Beispielsweise könnte ein übermäßiger Ladefehler, der auf ein Datenbankelement angewendet wird, eine Verbindungsanzahl als führenden Indikator haben. Sie können die stetige Zunahme der Verbindungsanzahl als führenden Indikator dafür sehen, dass die Datenbank das Verbindungslimit bald überschreitet, sodass Sie Maßnahmen ergreifen können, z. B. das Beenden der am wenigsten zuletzt verwendeten Verbindungen, um die Verbindungsanzahl zu reduzieren. Der Verzögerungsindikator gibt an, wann das Datenbankverbindungslimit überschritten wurde und

Datenbankverbindungsfehler zunehmen. Erwägen Sie nicht nur die Erfassung von Anwendungs- und Infrastrukturmetriken, sondern auch die Erfassung [von Key Performance Indicators \(KPI\)](#), um festzustellen, wann sich Ausfälle auf Ihr Kundenerlebnis auswirken.

Wenn möglich, empfehlen wir Ihnen, beide Arten von Indikatoren in Ihre Beobachtbarkeitsstrategie aufzunehmen. In einigen Fällen können Sie möglicherweise keine Frühindikatoren erstellen, sollten aber immer einen Verzögerungsindikator für jeden Fehler haben, den Sie beheben möchten. Um die richtige Abhilfe zu wählen, sollten Sie auch überlegen, ob ein Indikator mit oder ohne Verzögerung den Fehler erkannt hat. Betrachten Sie beispielsweise einen plötzlichen Anstieg des Datenverkehrs zu Ihrer Website. Wahrscheinlich wird nur ein Verzögerungsindikator angezeigt. In diesem Fall ist die automatische Skalierung allein möglicherweise nicht die beste Abhilfe, da es Zeit dauert, neue Ressourcen bereitzustellen, wohingegen eine Drosselung die Überlastung fast sofort verhindern und Ihrer Anwendung Zeit geben könnte, die Last zu skalieren oder zu reduzieren. Umgekehrt würden Sie bei einem allmählichen Anstieg des Datenverkehrs einen führenden Indikator sehen. In diesem Fall wäre eine Drosselung nicht angemessen, da Sie Zeit haben, durch automatische Skalierung Ihres Systems zu reagieren.

Allgemeine Strategien zur Abschwächung

Denken Sie zunächst über die Verwendung präventiver Abhilfemaßnahmen nach, um zu verhindern, dass sich der Fehlermodus auf die User History auswirkt. Dann sollten Sie über Korrekturmaßnahmen nachdenken. Korrekturmaßnahmen helfen dem System dabei, sich selbst zu erholen oder an sich ändernde Bedingungen anzupassen. Im Folgenden finden Sie eine Liste häufiger Abhilfemaßnahmen für jede Fehlerkategorie, die den Resilienzeigenschaften entspricht.

| Fehlerkategorie | Gewünschte Resilienz eigenschaften | Abschwächungen |
|-------------------------------|------------------------------------|--|
| Einzelne Fehlerpunkte (SPOFs) | Redundanz und Fehlertoleranz | <ul style="list-style-type: none"> • Implementieren Sie Redundanz – z. B. durch die Verwendung mehrerer EC2-Instances hinter Elastic Load Balancing (ELB). • Entfernen Sie Abhängigkeiten von der AWS globalen Service-Steuerebene und nehmen Sie Abhängigkeiten |

eiten nur von den globalen Service-Datenebenen auf.

- Verwenden Sie [eine ordnungsgemäße Verschlechterung](#), wenn eine Ressource nicht verfügbar ist, sodass Ihr System statisch stabil gegenüber einem einzelnen Ausfallpunkt ist.

Übermäßige Last

Ausreichende Kapazität

- Zu den wichtigsten Strategien zur Risikominderung gehören [Ratenbegrenzung](#), [Lasteinschränkung](#) und [Arbeitspriorität](#), [konstante Arbeit](#), [exponentielles Backoff](#) und [Wiederholung mit Jitter](#) oder gar kein Wiederholungsversuch, [Festlegung des kleineren Service in die Kontrolle](#), [Verwaltung der Warteschlangentiefe](#), [automatische Skalierung](#), [Vermeidung von Cold-Caches](#) und [Leistungsschaltern](#).
- Sie sollten auch Ihren Kapazitätsplan berücksichtigen und über zukünftige Kapazitäts- und Skalierungsgrenzen nachdenken, die sich sowohl auf AWS-Ressourcen als auch auf Limits in Ihrem System beziehen, die Sie möglicherweise erreichen.

Übermäßige Latenz

Zeitgesteuerte Ausgabe

- Implementieren Sie entsprechend konfigurierte [Timeouts](#) oder adaptive Timeouts (ändern Sie die Timeout-Werte basierend auf aktuellen und prognostizierten Latenzbedingungen, um möglicherweise eine langsame Abhängigkeit zu ermöglichen, Fortschritte zu machen, anstatt langsame Anfragen aufzugeben).
- Implementieren Sie [exponentielles Backoff und wiederholen Sie den Vorgang mit Jitter](#), Hedging und verwenden Sie Technologien wie [Multipath-TCP](#), wenn Sie von On-Premises-Umgebungen aus eine Verbindung zu Cloud-Services herstellen und Latenz über bestimmte Routen erfahren, indem Sie [asynchrone Interaktionen mit lose gekoppelten Systemen](#) verwenden, [zwischenspeichern und keine Arbeit verwerfen](#).

Fehlkonfiguration und Fehler

Korrekte Ausgabe

- Die primäre Möglichkeit, wiederholbare, funktionale Fehler in der Software zu erkennen, sind strenge Tests durch Mechanismen wie [statische Analysen](#), [Einheitentests](#), [Integrationstests](#), [Regressionstests](#), [Lasttests](#) und [Resilienztests](#).
- Implementieren Sie Strategien wie [Infrastructure as Code \(IaC\)](#) und [kontinuierliche Integration und kontinuierliche Bereitstellung \(CI/CD\)](#), um Bedrohungen durch Fehlkonfigurationen zu minimieren.
- Verwenden Sie Bereitstellungstechniken wie [One-Box-](#), [Canary-Bereitstellungen](#), Bruchteile Bereitstellungen, die auf Fehlerisolationsgrenzen ausgerichtet sind, oder [Blau/Grün-Bereitstellungen](#), um Fehlkonfigurationen und Fehler zu reduzieren.

Gemeinsam genutzte Datei

Fehlerisolierung

- Implementieren Sie die [Fehlertoleranz](#) in Ihrem System und verwenden Sie logische und physische Fehlerisolationsgrenzen wie mehrere Datenverarbeitungs- oder Container-Cluster, mehrere AWS-Konten, mehrere AWS Identity and Access Management (IAM)-Prinzipale, mehrere Availability Zones und möglicherweise mehrere AWS-Regionen.
- Techniken wie [zellenbasierte Architekturen](#) und [Shuffle Sharding](#) können auch die Fehlerisolierung verbessern.
- Berücksichtigen Sie Muster wie [lose Trichterung](#) und [ordnungsgemäße Verschlechterung](#), um kaskadierende Ausfälle zu vermeiden. Wenn Sie Benutzererfahrungen priorisieren, können Sie diese Priorisierung auch verwenden, um zwischen Benutzererfahrungen, die für die primäre Geschäftsfunktion unerlässlich sind, und Benutzererfahrungen zu unterscheiden, die ordnungsgemäß beeinträchtigt werden können. Auf

einer E-Commerce-Website möchten Sie beispielsweise nicht, dass sich eine Beeinträchtigung des Werbe-Widgets auf der Website auf die Fähigkeit auswirkt, neue Bestellungen zu verarbeiten.

Obwohl einige dieser Schutzmaßnahmen nur einen minimalen Aufwand für die Implementierung erfordern, könnten andere (z. B. die Einführung einer zellenbasierten Architektur für eine vorhersehbare Fehlerisolierung und minimale gemeinsame Ausfälle) eine Neugestaltung des gesamten Workloads und nicht nur der Komponenten einer bestimmten Benutzererfahrung erfordern. Wie bereits erwähnt, ist es wichtig, die Wahrscheinlichkeit und Auswirkungen des Fehlermodus gegen die Kompromisse abzuwägen, die Sie eingehen, um ihn zu minimieren.

Zusätzlich zu den Abschwächungstechniken, die für jede Fehlermodus-kategorie gelten, sollten Sie über Abschwächungen nachdenken, die für die Wiederherstellung der Benutzererfahrung oder des gesamten Systems erforderlich sind. Beispielsweise kann ein Fehler einen Workflow anhalten und verhindern, dass Daten in beabsichtigte Ziele geschrieben werden. In diesem Fall benötigen Sie möglicherweise operative Tools, um den Workflow erneut auszuführen oder die Daten manuell zu korrigieren. Möglicherweise müssen Sie auch einen Checkpointing-Mechanismus in Ihren Workload integrieren, um Datenverlust bei Ausfällen zu verhindern. Oder Sie müssen möglicherweise ein - und ein - und -Zeichen erstellen, um den Workflow zu unterbrechen und die Annahme neuer Arbeit zu beenden, um weitere Beeinträchtigungen zu vermeiden. In diesen Fällen sollten Sie über die betrieblichen Tools und Leitlinien nachdenken, die Sie benötigen.

Schließlich sollten Sie immer davon ausgehen, dass Menschen bei der Entwicklung Ihrer Abhilfestrategie Fehler machen werden. Obwohl moderne DevOps Methoden versuchen, den Betrieb zu automatisieren, müssen Menschen immer noch aus verschiedenen Gründen mit Ihren Workloads interagieren. Falsches menschliches Verhalten könnte zu einem Fehler in einer der SESpeed-Kategorien führen, z. B. zum Entfernen zu vieler Knoten während der Wartung und zum Verursachen einer Überlastung oder zum falschen Festlegen eines Feature-Flags. Diese Szenarien sind im Grunde ein Fehler bei präventiven Leitlinien. Eine Ursachenanalyse sollte niemals mit der Schlussfolgerung enden, dass „ein Mensch einen Fehler gemacht hat“. Stattdessen sollten die Gründe dafür berücksichtigt werden, warum Fehler von vornherein möglich waren. Daher sollte Ihre Abhilfestrategie überlegen, wie menschliche Operatoren mit Workload-Komponenten interagieren

können und wie Sie die Auswirkungen menschlicher Operatorfehler durch Sicherheitsleitlinien verhindern oder minimieren können.

Kontinuierliche Verbesserung

Ausfallsicherheit ist ein [kontinuierlicher Prozess](#). Während des Lebenszyklus Ihres Systems ändert sich die Umgebung, in der es arbeitet. Um sicherzustellen, dass Ihr System ausfallsicher bleibt, sollten Sie das Framework in Ihre regelmäßigen Betriebs- und Architekturüberprüfungen integrieren. Möglicherweise finden Sie neue Fehlermodi, die Sie beim ersten Mal nicht identifiziert haben, oder es kann neue oder bisher ungewollte Abhilfemaßnahmen geben, die Sie einrichten können. Die Resilienzanalyse sollte ein iterativer Prozess und keine einmalige Übung sein.

Sie sollten Ihre Strategien zur Abschwächung empirisch mit Prozessen [wiechaos-Engineering](#) oder [Spieltage](#) testen, um zu überprüfen, ob sie wie erwartet funktionieren. Wenn Sie keinen strengen Testmechanismus haben, sind Sie nicht sicher, dass die Abhilfe bei Bedarf wie erwartet funktioniert. Während der Resilienzanalyse können Sie feststellen, dass ein Fehlermodus bereits durch eine bestimmte Abhilfe behandelt wird, aber es ist wichtig, diese Annahmen ebenfalls zu testen. Sie sollten sowohl auf bestehende Abschwächungen als auch auf neue Abschwächungen testen, die mithilfe des Resilienzanalyse-Frameworks erstellt wurden.

Sie sollten auch anhand von Teamrückblicken bewerten, wie gut Sie die Analyse durchgeführt haben. Wussten alle, woran sie während der Analyse gearbeitet haben? Hat die Anzahl der Fehlermodi, die Sie durch die Resilienzanalyse gefunden haben, den Erwartungen des Teams entsprochen? Können Sie Abhilfemaßnahmen für alle erkannten Fehlermodi identifizieren? Hat das Team den Prozess nützlich gefunden? Glauben Sie, dass dies zu Verbesserungen der Ausfallsicherheit Ihres Workloads führen wird?

Wenn echte Fehlerereignisse auftreten, die sich auf die Verfügbarkeit Ihres Workloads auswirken, notieren Sie den spezifischen Fehlermodus, die Komponenten, die Teil des Fehlers waren, und das verwendete Abschwächungsmuster. Machen Sie diese Metadaten in Ihrem Analysetool nach Vorfällen durchsuchbar, damit Sie bestimmen können, auf welche Fehlermodi und Komponenten Sie sich in Zukunft konzentrieren möchten. Während dieses Prozesses können Sie Ihr AWS Kontoteam und Ihre Lösungsarchitekten einbeziehen.

Fazit und Ressourcen

Dieser Leitfaden bietet einen Rahmen für die kontinuierliche und konsistente Durchführung von Resilienzanalysen. Mithilfe dieses Frameworks können Sie ermitteln, wie sich einzelne Fehlerquellen, übermäßige Auslastung, übermäßige Latenz, Fehlkonfigurationen und Bugs sowie das gemeinsame Schicksal auf die Komponenten Ihres Workloads auswirken können. Die Identifizierung dieser Fehlermodi hilft Ihnen bei der Festlegung einer geeigneten Strategie zur Risikominderung als Teil des Aufbaus einer wiederherstellungsorientierten Architektur.

Weitere Informationen zur Resilienzanalyse finden Sie unter den folgenden Links:

- [Rahmen für den Lebenszyklus von Resilienz](#) (AWSPrescriptive Guidance)
- [Lösungen für Resilienz](#) (AWSLösungsbibliothek)
- [Auf dem Weg zu kontinuierlicher Resilienz](#) (Adrian Hornsby, The Cloud Architect, 24. März 2021)

Dokumentverlauf

In der folgenden Tabelle werden die wesentlichen Änderungen an diesem Leitfaden beschrieben. Wenn Sie über zukünftige Updates informiert werden möchten, können Sie ein abonnieren [RSS-Feed](#).

| Änderung | Beschreibung | Datum |
|--------------------------------------|--------------|-------------------|
| Erstveröffentlichung | — | 5. September 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 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 stressen, 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 unter [Kontinuierliche 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.

betreffene 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 Kerngeschäftsziele 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 Transaktionsdaten. 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.

IloT

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 Infrastrukturmgebung 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 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 kargestellt 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-Entitäts 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.

Refaktorisierung

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-Indicators](#).

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.