



Schnellere Beobachtbarkeitsergebnisse

## AWS Präskriptive Leitlinien



## AWS Präskriptive Leitlinien: Schnellere Beobachtbarkeitsergebnisse

Copyright © 2025 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
Zielgruppe .....	1
Ziele .....	2
Übersicht .....	3
Warum müssen Sie Ihre Beobachtbarkeitsstrategie überdenken .....	3
Tools und Frameworks für Observability .....	3
Stufe 1: Definiere deinen Nordstern .....	5
Integrieren Sie Observability zu einem früheren Zeitpunkt im Entwicklungszyklus (Shift-Left-Ansatz) .....	5
Richten Sie eine effektive Organisations- und Teamstruktur ein .....	7
Verfolgen Sie die Kostenverteilung .....	9
Definieren Sie Standards .....	9
Eskalationsprozesse einrichten .....	9
Verbessern Sie Ihre Fähigkeiten durch Schulungen .....	9
Stufe 2: Beobachtbarkeit implementieren .....	11
Wählen Sie Ihre Observability-Plattform .....	11
Funktionsumfang .....	11
Lizenzierungs- und Bereitstellungsmodell .....	12
Preisdimensionen .....	12
Fähigkeiten im Team .....	9
Betrieb und Wartung .....	13
Instrumentieren Sie Ihre Bewerbung .....	13
Erfassen Sie Telemetrie .....	14
Implementieren Sie Observability-Komponenten .....	14
Validieren Sie das Observability-System .....	15
Stufe 3: Inspizieren, anpassen und iterieren .....	16
Führen Sie regelmäßige Überprüfungen durch .....	16
Feiern Sie Siege .....	17
Lernen Sie aus Vorfällen .....	18
Nächste Schritte und Ressourcen .....	19
Ressourcen .....	19
Dokumentverlauf .....	20
Glossar .....	21
# .....	21

A .....	22
B .....	25
C .....	27
D .....	30
E .....	35
F .....	37
G .....	39
H .....	40
I .....	42
L .....	44
M .....	45
O .....	50
P .....	53
Q .....	56
R .....	56
S .....	59
T .....	63
U .....	65
V .....	66
W .....	66
Z .....	67
.....	lxix

# Schnellere Beobachtbarkeitsergebnisse

Jyothi Madanlal, Abhaya Chauhan und Jose Augusto Ferronato, Amazon Web Services

Juli 2025 ([Geschichte](#) der Dokumente)

Dr. Werner Vogels, Vice President und Chief Technology Officer bei Amazon, sagte einmal: „Alles schlägt fehl, die ganze Zeit... also plane für Misserfolge und nichts wird scheitern“ ([AWS re:Invent 2024 Keynote-Präsentation](#)). Die Übernahme dieser Denkweise steht im Mittelpunkt einer effektiven und robusten Implementierung von Observability.

In diesem Strategiedokument wird untersucht, wie Sie Ihre Beobachtbarkeit verbessern können, um widerstandsfähigere und effizientere Systeme zu schaffen. Es wird auch erörtert, wie Sie ein Ankerziel (North Star) definieren können, das für eine effektive Abstimmung innerhalb Ihres Unternehmens sorgt, was zu mehr Kundenvertrauen, Markenreputation und Teammoral führt. In dem Dokument werden die wichtigsten zu berücksichtigenden Aspekte und ein empfohlener Ansatz zur Beschleunigung der Beobachtbarkeitsergebnisse herausgestellt. Dieser Ansatz umfasst die Erschließung neuer Geschäftsmöglichkeiten, die Bereitstellung besserer Benutzererlebnisse, die Optimierung von Leistung und Kosten sowie die Verbesserung der betrieblichen Effizienz in Ihrem gesamten Unternehmen. Unabhängig vom Reifegrad Ihrer Observability-Implementierung bietet dieses Dokument schrittweise und umsetzbare Erkenntnisse zur Erreichung Ihrer Observability-Ziele.

## Zielgruppe

Diese Informationen richten sich an folgende Zielgruppen:

- Technische Führungskräfte und Architekten, die für die Definition einer Beobachtungsstrategie verantwortlich sind
- Technische Manager und DevOps Führungskräfte, die für die Überwachung der betrieblichen Exzellenz verantwortlich sind
- Plattform-Engineering-Teams, die für den Aufbau von Observability-Grundlagen verantwortlich sind
- Ingenieure für die Zuverlässigkeit von Websites (SREs), die für die Implementierung der Beobachtbarkeitspraktiken verantwortlich sind

## Ziele

Die Informationen in diesem Leitfaden helfen Ihnen dabei, Ihre Observability-Verfahren zu verbessern, um robustere und effizientere Systeme zu schaffen, die in Ihrem gesamten Unternehmen aufeinander abgestimmt sind. Es bietet Ihnen einen Rahmen für die Einführung guter Beobachtbarkeitspraktiken, die sich auf drei Säulen stützen: Mitarbeiter, Prozesse und Technologie. Die Informationen sind unabhängig von den Tools, die Sie verwenden, allgemein anwendbar. Wenn Sie an den AWS Diensten interessiert sind, die Beobachtbarkeit unterstützen, finden Sie weitere Informationen unter [Überwachung und Beobachtbarkeit](#) auf der AWS Website.

# Übersicht

## Warum müssen Sie Ihre Beobachtbarkeitsstrategie überdenken

Observability hat sich aus der Überwachung entwickelt, bei der der Schwerpunkt auf der Erfassung von Telemetriesignalen wie Protokollen, Metriken und Traces lag, um Sie beim Debuggen Ihrer Anwendungen zu unterstützen. Aufgrund dieses Zusammenhangs wurde Observability oft erst im Nachhinein berücksichtigt und führte zu zu viel oder zu wenig Instrumentierung, der Unfähigkeit, Signale zu korrelieren, unzusammenhängender Sichtbarkeit und einer Vielzahl von Tools, die oft nicht kohärent integriert wurden. Diese führten zu einem vermeintlichen Wertverlust und zu Kosten, die die Vorteile der Beobachtbarkeit zu überwiegen schienen. Aus geschäftlicher Sicht bedeuteten diese Probleme eine längere mittlere Erkennungszeit (MTTI), eine längere mittlere Wiederherstellungszeit (MTTR) und eine Verschlechterung der Benutzererfahrung, des Vertrauens, des Markenrufs und des Umsatzes. Bei Observability geht es heute nicht nur um die Fähigkeit, eine Anwendung zu debuggen und zu diagnostizieren, sondern auch um die Fähigkeit, zu überprüfen, ob sich die Anwendung genau wie beabsichtigt verhält.

Das Zusammentreffen von Unternehmen, die ihren Benutzern ein optimales Nutzererlebnis bieten wollen, und der Weiterentwicklung von Observability-Tools und -Funktionen erfordert ein Überdenken und eine Neupriorisierung der Beobachtbarkeit.

## Tools und Frameworks für Observability

Bevor sie 2019 auf den Markt OpenTelemetry kamen, machten spezielle Tools, die Observability-Lösungen für die Überwachung der Anwendungsleistung (APM) und die Überwachung digitaler Erlebnisse (DEM) bereitstellten, die Unterbrechungen zwischen den Telemetriesignalen sichtbarer und machten auf schlechte Benutzererfahrungen aufmerksam.

- APM verfolgt und analysiert das Verhalten von Softwareanwendungen in Echtzeit. Es misst wichtige Kennzahlen wie Reaktionszeiten, Fehlerraten und Ressourcennutzung und überwacht gleichzeitig Benutzertransaktionen für alle Anwendungskomponenten. APM-Tools helfen Teams dabei, Leistungsprobleme, Engpässe und Fehler schnell zu erkennen, bevor sich diese Probleme auf die Benutzer auswirken. Ihr primäres Ziel besteht darin, eine optimale Anwendungsleistung und Benutzererfahrung aufrechtzuerhalten und gleichzeitig den Zeitaufwand für die Problemlösung zu reduzieren.

- DEM misst und analysiert die Qualität der Interaktionen der Nutzer mit digitalen Diensten aus ihrer Sicht. Es kombiniert reale Benutzerüberwachung (RUM), synthetische Überwachung und Endpunktüberwachung, um einen vollständigen Überblick über die Benutzererfahrung zu bieten. DEM verfolgt Metriken wie Seitenladezeiten, Reaktionsfähigkeit von Anwendungen und den Abschluss der Benutzererfahrung auf verschiedenen Geräten, Browsern und Standorten. Dies hilft Unternehmen zu verstehen, wie Benutzer ihre digitalen Dienste erleben, Leistungsprobleme zu identifizieren, die sich auf die Benutzerzufriedenheit auswirken, und digitale Kontaktpunkte zu optimieren. Die Erkenntnisse ermöglichen es Unternehmen, datengestützte Entscheidungen zu treffen, um das Kundenerlebnis zu verbessern und Wettbewerbsvorteile zu wahren.

Mit der Einführung von [OpenTelemetry](#) im Jahr 2019 wurde ein einheitlicher Open-Source-Standard für die Generierung, Erfassung, Verwaltung und den Export von Telemetriedaten bereitgestellt. Dieses Framework konzentriert sich darauf, die Lücken zwischen Telemetriesignalen zu schließen, indem es Kontext hinzufügt, eine bessere Korrelation zwischen den Signalen bietet und einen besseren abgeleiteten Wert bietet. Die Verwendung strukturierter Protokolle mit zusätzlichem Kontext hilft Ihnen beispielsweise dabei, Metriken aus den aufgenommenen Protokollen abzuleiten und die Informationen auf unterschiedliche Weise zu analysieren, um schneller zur Ursache zu gelangen. Bisher OpenTelemetry wurden Signale isoliert betrachtet. Um Funktionen hinzuzufügen, mussten Sie den Code überarbeiten, um einer vorhandenen Metrik eine neue Dimension hinzuzufügen oder eine neue Metrik zu erstellen. Sie mussten warten, bis der Code den Entwicklungszyklus durchlief, und dann warten, bis die Metriken in einer geeigneten Umgebung beobachtet wurden, bevor Sie Schlussfolgerungen ziehen konnten. Dieser Prozess verzögerte die Sichtbarkeit und beeinträchtigte Ihre Fähigkeit, die Daten gegebenenfalls mit Protokollen oder Traces zu korrelieren.

Der Support für OpenTelemetry und die daraus resultierenden Tool-Verbesserungen helfen Ihnen dabei, mehr Nutzen aus Observability-Plattformen zu ziehen, die Benutzererfahrung zu verbessern und sowohl die betriebliche Effizienz als auch die Arbeitsmoral zu verbessern.

Wenn Sie Ihre Haltung zur Beobachtbarkeit verbessern und verbessern möchten, wo und wie fangen Sie dann eigentlich an? Wir empfehlen einen Ansatz, der aus drei Schritten besteht, die in diesem Leitfaden ausführlich behandelt werden:

- [Stufe 1: Definiere deinen Nordstern](#)
- [Phase 2: Implementieren Sie Beobachtbarkeit](#)
- [Phase 3: Untersuchen, Anpassen und Iterieren](#)

# Stufe 1: Definiere deinen Nordstern

Bei einer erfolgreichen Implementierung von Observability geht es nicht nur um Abläufe und Tools — es geht auch darum, eine Kultur der Eigenverantwortung, der kontinuierlichen Verbesserung und der proaktiven Problemlösung zu fördern. Wie bei jeder erfolgreichen Strategie erfordert Ihre Strategie für Observability eine ganzheitliche Betrachtung von drei Säulen: Mitarbeiter, Prozesse und Technologie.

Wenn Sie Ihre Observability-Haltung etablieren oder verbessern möchten, empfehlen wir Ihnen, zunächst zu definieren, worauf es ankommt, von Ihren Geschäftsergebnissen auszugehen und Ihre Strategie kontinuierlich zu überprüfen, anzupassen und neu auszurichten, wenn sich Ihr Unternehmen, Ihre Teams und Produkte weiterentwickeln.

In dieser ersten Phase definieren und etablieren Sie Ihren North Star. Dabei handelt es sich um eine vereinbarte und gut verstandene Definition dessen, was gut für Ihr Unternehmen aussieht. Wir empfehlen Ihnen, einige oder alle Aktivitäten in dieser Phase erneut zu überprüfen, wenn sich Ihr Unternehmen weiterentwickelt, wenn Sie ein neues Produkt, eine neue Anwendung oder einen neuen Service auf den Markt bringen oder wenn Sie eine größere architektonische Änderung planen, um Ihre Observability-Plattform und Ihre organisatorischen Anforderungen neu zu bewerten.

## Integrieren Sie Observability zu einem früheren Zeitpunkt im Entwicklungszyklus (Shift-Left-Ansatz)

Machen Sie Observability zu einer Verantwortung für jedes Mitglied der Technik-, Betriebs- und Produktteams und behandeln Sie sie als primäre funktionale Anforderung, ähnlich wie Sie Unit-Tests oder Sicherheit behandeln. Dadurch wird die Verantwortung nicht vom Betriebsteam auf das Entwicklungsteam verlagert, sondern die erforderliche Zusammenarbeit zwischen den verschiedenen Teams hervorgehoben. Für Teams ist es hilfreich, zu Beginn des Entwicklungszyklus die folgenden Aktivitäten gemeinsam durchzuführen. Möglicherweise möchten Sie diese pro Ticket, pro Funktion oder pro Produkt durchführen.

- Identifizieren Sie die Beteiligten. Wer sind die Stakeholder und was ist für sie wichtig, wenn diese Funktion oder dieses Produkt nicht wie erwartet funktioniert? Berücksichtigen Sie bei der Identifizierung der Beteiligten Aspekte wie Funktionalität, Verfügbarkeit, Sicherheit, Kosten, Vertrieb und Produktnutzung. Zu den Stakeholdern können Ihr Team, die Kunden Ihres Produkts, interne Geschäftsbeteiligte, Mitglieder des Plattformbetriebsteams und Anwendungsentwickler gehören. Je nach Szenario können auch Ihre Sicherheits- und Finanzteams Interessengruppen sein.

- Identifizieren Sie die wichtigsten Ergebnisse. Ermitteln Sie die wichtigsten Ergebnisse und ihre Auswirkungen auf das Unternehmen und die einzelnen Stakeholder. Identifizieren Sie Erfolg und Misserfolg für jedes Ergebnis und jeden Stakeholder. Die Ergebnisse werden in der Regel als Leistungsziele (SLOs) definiert und müssen quantifizierbar sein. Ein SLO ist ein Maß für jedes Ergebnis. Ein gutes SLO hat einen Zielwert, der als Ziel angestrebt oder beibehalten werden sollte. Ein SLO kann ein Maß für die Benutzerzufriedenheit sein. Ein Service Level Indicator (SLI) ist der tatsächliche Messwert oder die Metriken, anhand derer bestimmt wird, ob Sie den SLO erfüllen. Es ist der quantifizierbare Datenpunkt, den Sie anhand Ihres Ziels verfolgen. Beispiele hierfür sind die Reduzierung der MTTR um 60 Prozent, die Aufrechterhaltung der Anwendungsverfügbarkeit bei 99,99 Prozent oder die Verbesserung der Entwicklerproduktivität um 30 Prozent.

Nehmen wir das Beispiel der Aufrechterhaltung der Anwendungsverfügbarkeit bei 99,99 Prozent und definieren Sie die SLO-, SLI- und Metriken, die zur Erfolgsmessung und Validierung erforderlich sind. Betrachten wir für dieses Beispiel eine RESTful Anwendung und definieren Anwendungsverfügbarkeit als den erfolgreichen Abschluss aller eingehenden Anfragen. Dazu müssen die Gesamtzahl der Anfragen an die Anwendung und der Abschlussstatus jeder Anfrage gemessen werden. Wenn Sie diese in SLO und SLI übersetzen, benötigen Sie eine Metrik, die eingehende Anfragen erfasst, und eine weitere Metrik, die den Status von Anfragen erfasst. Wenn alle Anfragen erfolgreich abgeschlossen wurden, gilt die Anwendung als verfügbar. Wenn eine oder mehrere Anfragen zu Fehlern führen, gilt die Anwendung als nicht verfügbar. Der SLI wäre also die Summe der fehlerhaft abgeschlossenen Anfragen geteilt durch die Summe der eingehenden Anfragen in einem 5-Minuten-Intervall — das entspricht einer Fehlerquote. Sie können diesem SLI ein Ziel hinzufügen, um daraus ein SLO zu machen. Beispiel: Bemühen Sie sich, dass die Fehlerquote in 3 aufeinanderfolgenden 5-Minuten-Intervallen unter 0,1 Prozent liegt.

- Priorisieren Sie wichtige Ergebnisse. Basierend auf der Priorität, die Sie für jedes Ergebnis festlegen, können Sie sich dafür entscheiden, sich zuerst auf die Ergebnisse zu konzentrieren, die die größte Wirkung haben, anstatt alles gleichzeitig zu tun. Fangen Sie klein an, wiederholen Sie und verbessern Sie Ihre Beobachtbarkeit in kleinen Schritten. Observability ist ein Prozess, der fortlaufende Überprüfungen, Audits, Verbesserungen und Verbesserungen erfordert, um den Reifegrad und die Vorteile zu erhöhen. Die Priorisierung bietet Ihnen auch die Möglichkeit, schrittweise Meilensteine zu definieren, um die ermittelten Ergebnisse zu erreichen.
- Identifizieren Sie die erforderliche Instrumentierung. Welche Komponenten und zugehörigen Merkmale der Architektur oder Implementierung können die Ergebnisse beeinflussen, auf die es ankommt, wie in den vorherigen Schritten beschrieben? Wenn Sie beispielsweise eine Anwendung auf einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance ausführen, können sich die Anzahl der Kerne und der verfügbare Arbeitsspeicher auf die Reaktionsfähigkeit und den

Durchsatz der Anwendung auswirken. In dieser Phase kann es auch hilfreich sein, festzustellen, ob die Tools oder Bibliotheken, die Sie verwenden, bereits einen Teil dieser Instrumentierung bereitstellen. Durch die Durchführung einer Reihe von Vorprüfungen oder das Hinzufügen von Fragen wie den folgenden zur [Definition von bereit \(DoR\)](#) eines Tickets kann diese Aktivität Teil des Standardprozesses werden.

- Was müssten Sie wissen, um den Fehler zu beheben, falls dieser Vorgang fehlschlagen sollte? Wie wirkt sich ein typischer oder problematischer Vorgang auf die beteiligten Komponenten aus? Welche Art von Signal sollte dieser Vorgang senden: Protokoll, Metrik oder Trace? Wie hoch sind die Kosten dieser Instrumente im Vergleich zu ihrem Wert? Welche Art von Aggregation wäre ohne Sicherheitsverletzung akzeptabel? SLOs
- Welche Komponenten und Abhängigkeiten können zu einem Ausfall dieses Vorgangs führen? Wie werden Sie feststellen, welche Komponente oder Abhängigkeit den Fehler verursacht hat? Was sind die verschiedenen Konfigurationshebel dieser Komponenten und Abhängigkeiten, und wie wirken sie sich jeweils auf den Betrieb aus?
- Welche metrische Granularität und Abtastrate sind erforderlich, um sicherzustellen, dass SLI und SLO genau gemessen werden können?
- Definieren Sie Erfolgskriterien. Definieren Sie für jedes priorisierte Ergebnis Schwellenwerte, die auf die Auswirkungen des Erreichens oder Nichterreichen von Zielen abgestimmt sind. Die Erfolgskriterien bieten Teams zusätzlichen Kontext, wenn sie auf Warnmeldungen reagieren. Sie bieten Ihnen auch die Möglichkeit, Prognosen zu erstellen und Kompromisse mit den Kosten der Instrumentierung einzugehen, um die erforderliche Transparenz zu erreichen.

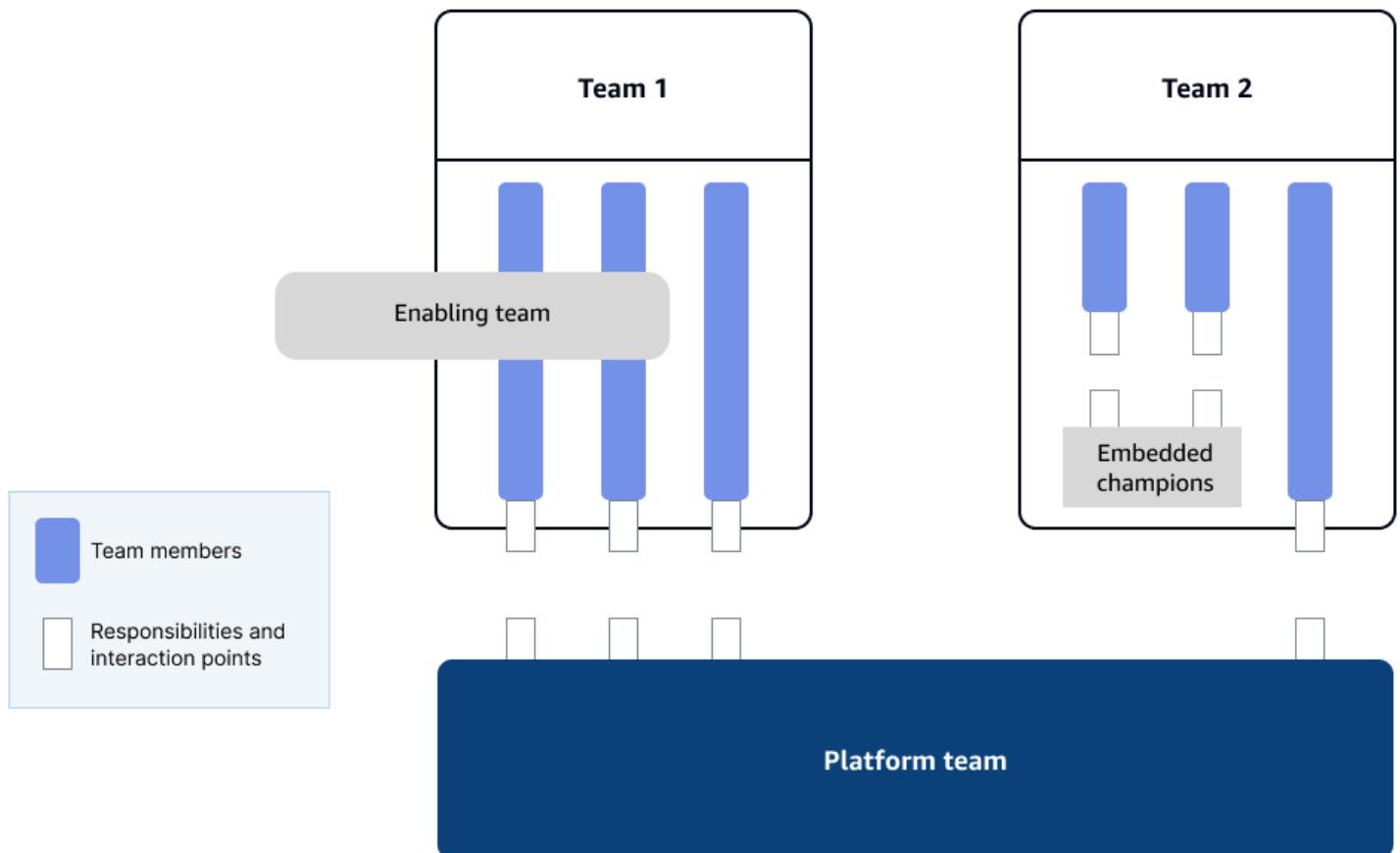
## Richten Sie eine effektive Organisations- und Teamstruktur ein

Je nach architektonischer Komplexität und Größe Ihres Unternehmens müssen Sie möglicherweise ein eigenes Team zusammenstellen, das sich auf die Beobachtbarkeit konzentriert. Dieses Team wird für die Konfiguration der Observability-Tools und die Einrichtung der Observability-Plattform für andere Teams verantwortlich sein. Wir empfehlen außerdem, ein engagiertes Team einzurichten, wenn Sie sich für eine OpenTelemetry Standardimplementierung entscheiden. In kleineren Organisationen können Sie jedem Teammitglied Observability als zusätzliche Verantwortung zuweisen und auch Observability-Champions ernennen, die sich für die Verbreitung und Durchsetzung von Best Practices in allen Teams einsetzen. Diese Champions engagieren sich ehrenamtlich für einen Teil ihres Tages, um Prozesse zu definieren und Standards für die Organisation festzulegen. Sie arbeiten entweder als Team, das sich selbst normalisiert, oder sie

können von engagierten Observability-Experten geleitet werden. Das folgende Diagramm zeigt, wie Ihre Investition Ihren organisatorischen Ansatz bestimmen kann.



Die Champions könnten vollständig in Teams integriert sein (wie in der folgenden Abbildung für Team 2 gezeigt) oder Teil eines Teams sein, das sich zwischen den Teams abwechselt, um bewährte Verfahren festzulegen und zu fördern (Team 1 in der Abbildung).



## Verfolgen Sie die Kostenverteilung

Organizations sollten eine umfassende Kostenverfolgung und Transparenz in Bezug auf Kennzahlen, Protokolle und Traces implementieren und gleichzeitig eine teamspezifische Rechenschaftspflicht für den Ressourcenverbrauch und die Kosten einrichten. Für eine erfolgreiche Integration von Finanzoperationen (FinOps) sind automatisierte Überwachungssysteme mit Budgetwarnungen erforderlich, die mit einer systematischen Datenspeicherung und Optimierung der Datenerfassung einhergehen. Die Ingenieur- und Finanzteams sollten ihre Ziele durch gemeinsame Dashboards und regelmäßige Überprüfungen aufeinander abstimmen. Organizations profitieren von der Implementierung klarer Chargeback-Modelle und Strategien zur Kostenverteilung, um Eigenverantwortung und Rechenschaftspflicht zu fördern.

## Definieren Sie Standards

Identifizieren und definieren Sie die Basissignale und die Telemetrie, die eine Anwendung benötigt, einschließlich Alarmierungs- und Dashboard-Strategien. Erstellen Sie für jeden Antrag eine Checkliste oder einen formellen Überprüfungsprozess. Auf der Website „[Bewährte Methoden für AWS Observability](#)“ finden Sie Richtlinien für Benachrichtigungen und die Erstellung von Dashboards, z. B. die Festlegung geeigneter Schwellenwerte für Warnmeldungen, die Minimierung der Warnungsmüdigkeit, die Erstellung von Dashboards mit ausreichend Kontext für jede Person usw. Informationen zu vernetzten und kuratierten Observability-Erlebnissen finden Sie unter [Anwendungssignale](#) in der CloudWatch Amazon-Dokumentation.

## Eskalationsprozesse einrichten

Es ist wichtig, Eskalationsmechanismen, Eigenverantwortung für Warnmeldungen und Reaktionsverfahren einzurichten und durchzusetzen. Wir empfehlen Ihnen, eine Kultur zu fördern, in der Eskalation nicht verpönt ist.

## Verbessern Sie Ihre Fähigkeiten durch Schulungen

Finden Sie heraus, wie Sie bestehende und neue Teammitglieder am besten weiterbilden können, unterstreichen Sie die Bedeutung der Beobachtbarkeit und fördern Sie eine Kultur der kontinuierlichen Verbesserung. Je nach den Bedürfnissen Ihres Unternehmens können Sie zwischen aufgezeichneten On-Demand-Schulungen oder Präsenzschulungen wählen, die von Experten oder Spezialisten im Bereich Observability durchgeführt werden. Ihr AWS-Konto Team kann ausführliche,

praktische Schulungen wie den [One Observability Workshop](#) oder [GameDaysSchulungen](#) anbieten, in denen Sie Ihre Fähigkeiten im Bereich Observability trainieren und Ihre Fähigkeiten und Best Practices verbessern können. Integrieren Sie außerdem Mechanismen zur Stärkung bewährter Verfahren und zur Förderung der von Ihrer Organisation festgelegten Standards.

# Stufe 2: Beobachtbarkeit implementieren

In dieser Phase beginnen Sie mit dem Prozess, damit sich Ihre Teams schrittweise bis zum North Star vorarbeiten können.

## Wählen Sie Ihre Observability-Plattform

Der erste Schritt besteht darin, die richtigen Tools zu finden, um die Signale aufzunehmen, zu visualisieren und zu analysieren und Warnmeldungen zu senden. Bei der Auswahl eines Tools sollten Sie dessen Funktionsumfang, Lizenzmodell, Preis, Qualifikationsanforderungen und Wartung berücksichtigen.

### Funktionsumfang

Hier sind einige der Fragen, die Sie berücksichtigen sollten:

- Konfigurierbarkeit und Anpassung. Welche Funktionen bietet das Tool, um die Ermittlungserfahrung zu vereinfachen und die MTTR zu verkürzen? Bietet das Tool Alarmkorrelation, metrische Mathematik, Flexibilität beim Umgang mit fehlender Telemetrie oder Erkennung von Anomalien?
- Granularität. Welche Granularität wird bei der Erfassung und Visualisierung von Telemetriesignalen unterstützt?
- Personas. Unterstützt das Tool die Erlebnisse, die Sie Ihren Entwicklern, Plattformtechnikern und anderen Personas bieten möchten? Funktioniert es sowohl für technische als auch für Geschäftsleute?
- Widgets. Welche Arten von Widgets unterstützen die Dashboards? Erlaubt das Tool die Erstellung von benutzerdefinierten Widgets?
- Vorgefertigte Lösungen. Welche vorgefertigten Observability-Lösungen bietet das Tool, um die Amortisationszeit zu verkürzen?
- Automatisierung und generative KI. Welche Funktionen bietet das Tool, mit denen Sie und Ihr Team den Arbeitsaufwand automatisieren oder reduzieren können? Beispielsweise können automatische Anomalieerkennung, prädiktive Analysen und andere generative KI-Funktionen dazu beitragen, den Stress von Annahmen und Unbekannt-Unbekannten (also Dingen, die Sie weder kennen noch vollständig verstehen) zu reduzieren. Unterstützt das Tool die Verwendung generativer AI/ML Modelle, um die Analyse der Daten im großen Maßstab zu verbessern? Bietet es Ihnen die Möglichkeit zur Automatisierung und Implementierung AIOps?

- Sicherheit. Welche Arten von Authentifizierungs- und Autorisierungsintegrationen unterstützt das Tool? Entsprechen die Benutzer- und Anmeldemöglichkeiten den Anforderungen Ihres Unternehmens?
- OpenTelemetry Unterstützung. Unterstützen das Tool und der Agent OpenTelemetry? Die meisten Observability-Plattformen unterstützen die Aufnahme von OpenTelemetry -kompatiblen Signalen, aber nicht alle Agenten bieten Konfigurationsoptionen, um diese Signale an eine Observability-Plattform weiterzuleiten.
- Integrationen. Welche Integrationen bietet das Tool? Überlege dir, ob du Benachrichtigungen an Slack senden, Teammitglieder kontaktieren oder die Problembehebung automatisieren möchtest.
- Skalierbarkeit. Wie skalierbar und performant ist das Tool? Die Observability-Lösung muss mit Ihren Anforderungen und der steigenden Nutzung skaliert werden, sodass sie auch dann Diagnosen stellen kann, wenn Ihre Anwendung nicht verfügbar ist.
- Support. Welches Support-Modell wird angeboten? Ihr Observability-Tool muss auch dann verfügbar sein, wenn Ihre Anwendung ausfällt, damit Sie Ihre MTTR- und Anwendungsverfügbarkeitsziele oder Service Level Agreements () einhalten können. SLAs Open-Source-Lösungen bieten möglicherweise nur begrenzten formellen Support.

## Lizenzierungs- und Bereitstellungsmodell

Berücksichtigen Sie das Lizenzmodell (Open Source oder kommerziell) und das Bereitstellungsmodell (selbst gehostet oder cloudbasiert) der Lösung. Open-Source-Optionen haben oft geringere Vorlaufkosten, erfordern jedoch möglicherweise mehr Zeit für Bereitstellung, Einrichtung und Konfiguration, Wartung und Teamqualifizierung, bevor sie einen Mehrwert bieten. Wenn Sie Open-Source-Optionen in Betracht ziehen, benötigen Sie möglicherweise ein engagiertes Team von Observability-Experten. Kommerzielle Software bietet in der Regel eine kürzere Amortisierungszeit bei höheren Vorlaufkosten, und der Bedarf an einem eigenen Observability-Team wächst mit der Zeit, wenn Akzeptanz, Komplexität und Reife zunehmen. Selbst gehostete Lösungen erfordern im Vergleich zu cloudbasierten Lösungen mehr Zeit für Bereitstellung, Einrichtung und Konfiguration, Wartung und Betriebskosten.

## Preisdimensionen

Wie wird sich das Preismodell des Tools auf Ihre Gesamtbetriebskosten (TCO) auswirken, wenn Ihre Anwendung neue Benutzer gewinnt, die Architektur größer wird oder neue Funktionen und Anwendungen hinzukommen? Einige typische Lizenzmodelle sind beispielsweise unbefristet oder basieren auf Abonnements, der Anzahl der benannten Benutzer, dem Verbrauch oder dem Volumen.

Überlegen Sie, wie Ihre Anwendung und das Observability-Tool im Hinblick auf die Nutzung skalieren werden und wie sich das Lizenzmodell auf die Kosten des Tools auswirken kann.

## Fähigkeiten im Team

Abhängig von den aktuellen Fähigkeiten und dem Reifegrad Ihres Teams müssen Sie festlegen, wie viel Weiterbildung erforderlich sein wird. Überlegen Sie, welche Art von Unterstützung der Anbieter für die Schulung Ihres Teams anbietet. Überlegen Sie auch, ob Ihre Organisationsstruktur die Konfiguration und Verwaltung des von Ihnen ausgewählten Tools unterstützt. Wenn Sie sich beispielsweise dafür entscheiden OpenTelemetry, sollten Sie in Betracht ziehen, ein eigenes Team zusammenzustellen, das sich auf Beobachtbarkeit spezialisiert hat.

## Betrieb und Wartung

Beurteilen Sie die folgenden Fragen:

- Welche Bereitstellungsoptionen bietet der Observability Agent oder Collector? Entsprechen diese Optionen den Anforderungen Ihrer Architektur? Wenn Sie beispielsweise eine containerisierte Bereitstellung für das Observability-Tool verwenden, unterstützt es dann ein Daemonset oder Sidecar? Welche zusätzlichen Schritte oder Tools müsste das Betriebsteam ergreifen oder einsetzen, um sicherzustellen, dass die Sicherheit und alle anderen Prozesse eingehalten werden?
- Welcher Aufwand ist für die Wartung der Lösung erforderlich? Wie einfach oder automatisiert ist der Prozess der Aktualisierung des Agenten oder Collectors? Vollständig verwaltete und cloudbasierte Observability-Schnittstellen haben im Vergleich zu selbst bereitgestellten und gehosteten Anwendungen in der Regel einen geringeren Betriebsaufwand, obwohl die Verwaltung des Agenten oder Collectors gleich bleibt. Berücksichtigen Sie Ihre Teamstruktur und berücksichtigen Sie die Personalkosten, die mit der Wartung der von Ihnen ausgewählten Option verbunden sind.

## Instrumentieren Sie Ihre Bewerbung

Die Antworten auf die Fragen im vorherigen Abschnitt geben Ihnen die Informationen, die Sie für die Instrumentierung Ihrer Anwendung benötigen, d. h., um Ihrer Anwendung Code zur Erfassung von Telemetriesignalen hinzuzufügen und Verhaltensweisen zu messen, zu beobachten und zu validieren. Sie können SDKs beispielsweise das OpenTelemetry SDK für die Sprache Ihrer Anwendung verwenden, um Ihre Anwendung automatisch zu instrumentieren. Möglicherweise müssen Sie dennoch manuellen Instrumentierungscode hinzufügen, um etwaige Lücken zu schließen

und die end-to-end Sichtbarkeit zu gewährleisten. Gehen Sie bewusst mit der Telemetrie um, die Sie hinzufügen, und stellen Sie sicher, dass Sie sie wieder mit einer oder mehreren SLIs Telemetriedaten verbinden können SLOs, die Sie in der vorherigen Phase eingerichtet haben.

## Erfassen Sie Telemetrie

Konfigurieren Sie den Telemetrie-Collector oder -Agenten so, dass die relevanten Telemetriesignale entsprechend den Ergebnissen, die Sie in Phase 1 priorisiert haben, aufgenommen werden.

## Implementieren Sie Observability-Komponenten

Wenn die Telemetrie fließt und in eine Observability-Plattform aufgenommen wird, erstellen Sie Dashboards, Warnmeldungen, Playbooks und Runbooks.

- Dashboards: Erstellen Sie Dashboards, die relevante Informationen enthalten, einschließlich einer visuellen Darstellung aktueller und historischer Trends im Zusammenhang mit Ihren priorisierten Ergebnissen. Stellen Sie diese Dashboards den Stakeholdern zur Verfügung, die Sie in Phase 1 definiert haben. Weitere Informationen finden Sie unter Aufbau von Dashboards für betriebliche Transparenz auf der Website der Amazon Builders' Library.
- Warnmeldungen: Definieren Sie Warnmeldungen, um Ihr Team zu benachrichtigen, wenn die Ergebnisse gefährdet sind oder verletzt werden. Erwägen Sie, Warnmeldungen für Sicherheits- und Leistungsprobleme hinzuzufügen. Optimieren Sie Warnmeldungen, um die Warnungsmüdigkeit und die damit verbundenen Kosten zu reduzieren, indem Sie folgende Maßnahmen ergreifen:
  - Verwenden Sie die Anomalieerkennung, um die Festlegung fester Schwellenwerte zu vermeiden, die häufig angepasst werden müssen, und um das Auftreten von Unbekannt-Unbekannten zu reduzieren.
  - Verwenden Sie intelligente Alert-Kombinationen, die mehrere verwandte Metriken zusammen betrachten, anstatt einzelne Alerts für jede Metrik einzurichten. Anstatt beispielsweise separate Warnmeldungen für CPU, Arbeitsspeicher und Reaktionszeit einzurichten, sollten Sie eine aussagekräftige Warnung erstellen, die nur ausgelöst wird, wenn diese Metriken zusammen auf ein echtes Problem hinweisen. Dieser Ansatz reduziert die Anzahl der Warnmeldungen erheblich und hilft Ihren Teams, sich auf tatsächliche Probleme zu konzentrieren, die sich auf den Service auswirken, anstatt auf einzelne Messspitzen reagieren zu müssen.
  - Generieren Sie Warnmeldungen nur, wenn die Benutzererfahrung oder die Ergebnisse gefährdet sind. Vermeiden Sie es beispielsweise, Benachrichtigungen über eine CPU-Spitze zu erhalten,

die durch ein automatisiertes Upgrade verursacht wird, wenn Ihre Anwendung keine aktiven Benutzer hat.

- Playbooks: Ein Playbook bietet der Person, die auf einen Vorfall oder eine Warnung reagiert, ein mentales Modell und einen Kontext und hilft ihr, die Ursache schneller zu identifizieren. [Erwägen Sie die Erstellung von Playbooks für hochgradig gekoppelte, komplexe Anwendungen und für Anwendungen, denen es an Instrumentierung mangelt, die sich jedoch direkt auf die Ergebnisse auswirken, die Sie in Phase 1 identifiziert und priorisiert haben.](#)
- Runbooks: Verwenden Sie Runbooks, um die Schritte zu definieren, die zur Behebung eines Vorfalls oder einer Warnung erforderlich sind.

## Validieren Sie das Observability-System

Stellen Sie während Ihres gesamten Softwareentwicklungszyklus (SDLC) sicher, dass die Dashboards die erwarteten Verhaltensweisen und Aktualisierungen während der Systemtests bieten. Implementieren Sie [Chaos Engineering](#) und validieren Sie die in Playbooks und Runbooks dokumentierten Schritte, um sicherzustellen, dass sie korrekt sind und ihren Zweck erfüllen. Sie sollten auch die Inhaberschaft der Warnmeldungen und die Eskalationswege überprüfen.

## Stufe 3: Inspizieren, anpassen und iterieren

Nach der Implementierung Ihres Observability-Systems empfehlen wir Ihnen, Ihre Implementierung kontinuierlich zu überprüfen, zu bewerten, zu erlernen, anzupassen und zu verbessern. Sie können das [AWS Observability Maturity Model](#) als Tool verwenden, um den Reifegrad Ihrer Implementierung zu bewerten und Bereiche mit Verbesserungspotenzial zu identifizieren und zu priorisieren.

### Führen Sie regelmäßige Überprüfungen durch

Beobachtbarkeit ist ein iterativer Prozess. Es erfordert regelmäßige Audits und Bewertungen vorhandener Komponenten sowie Änderungen und Verbesserungen, um eine kontinuierliche Verbesserung voranzutreiben. Wir empfehlen Ihnen, regelmäßige Überprüfungen durchzuführen, um Schwellenwerte für Warnmeldungen SLOs, Dashboards, Metrikgranularität, Aufbewahrungsrichtlinien, Stichprobenstrategien usw. neu zu bewerten, um sicherzustellen, dass diese den Mehrwert für Ihre Teams und Ihr Unternehmen steigern. Indem Sie die Kosten für die Beobachtbarkeit mit bestimmten Teams und Services verknüpfen, können Sie datengestützte Entscheidungen über den Umfang und die Ressourcenzuweisung treffen.

Bei Amazon führen wir wöchentliche [Operational Readiness Readiness Reviews \(ORRs\)](#) durch, um die Prozesse und den Zustand der Beobachtbarkeit der Teams anhand bewährter Verfahren zu überprüfen. Dabei handelt es sich um eine nicht blockierende Maßnahme, die der Anzahl der Services und der Häufigkeit der Veröffentlichungen bei Amazon entspricht.

Je nach Größe Ihres Unternehmens können Sie auch einen Dienstplan wie gewohnt (BAU) einrichten, bei dem ein Mitglied jedes Teams dafür verantwortlich ist, über Anomalien und Trends zu berichten, unbekannte Unbekannte aufzudecken, unerwünschte Instrumente und Benachrichtigungen zu entfernen, Dashboards zu verbessern und sicherzustellen, dass die Observability-Lösung weiterhin für das Team funktioniert und auf die Ziele und Erfolgskennzahlen des Teams abgestimmt ist. Dies könnte auch eine Gelegenheit sein, die Warnstrategie zu überdenken, um schneller, proaktiver und näher am Benutzer zu sein. [Ziel dieser Überprüfungen ist es, einen positiven Kreislauf zu schaffen, wie in der folgenden Abbildung dargestellt, und den Reifegrad Ihrer Observability-Haltung zu verbessern, wie im Observability Maturity Model beschrieben.](#) AWS



Identifizieren Sie die Playbooks, auf die am häufigsten zugegriffen wird, und erwägen Sie, Ihre Anwendung zu verbessern oder weitere Instrumente hinzuzufügen. Identifizieren Sie die Runbooks, die am häufigsten ausgeführt werden, und erwägen Sie, diese Runbooks zu automatisieren.

Die Erkenntnisse aus diesen Überprüfungen werden auch dem Observability-Team und den Experten zur Verfügung gestellt, um Verbesserungen der zentralen Programme und der Observability-Plattform aufzuzeigen. Abhängig von der Häufigkeit der durch die Bereitstellung ausgelösten Ereignisse könnten Sie sich beispielsweise dafür entscheiden, der Verbesserung der Bereitstellungspipeline Vorrang vor anderen Komponenten einzuräumen. Wenn die MTTR aufgrund von Überwachungslücken höher ist, können Sie der Verbesserung der Observability-Plattform und ihrer Konfiguration Priorität einräumen.

## Feiern Sie Siege

Erzählen Sie Erfolgsgeschichten von Teams, die Observability-Tools verwenden. Heben Sie beispielsweise den Erfolg eines Teams hervor, das anhand von Beobachtungskennzahlen eine alternative Lösung implementiert hat, die effizienter ist und zu einer geringeren Latenz oder geringeren Kosten führt. Die Kommunikation dieses Erfolgs unterstreicht die Bedeutung der

Beobachtbarkeit und motiviert andere Teams, ihre Haltung zur Beobachtbarkeit zu verbessern und ähnliche Erfolge anzustreben.

## Lernen Sie aus Vorfällen

Führen Sie nach einem Vorfall untadelige Übungen durch, die dem Prozess zur Fehlerkorrektur (COE) bei Amazon ähneln, um Bereiche mit Verbesserungspotenzial zu identifizieren und future Probleme zu vermeiden. Wie bei Siegen können die Erkenntnisse aus dieser Übung umfassend mit anderen Teams geteilt werden, um den Wert von Beobachtbarkeit und bewährten Verfahren zu stärken.

# Nächste Schritte und Ressourcen

Wenn Sie mit Ihren Benutzern und Geschäftsergebnissen beginnen und sich auf die Auswirkungen konzentrieren, können Sie einen größeren Nutzen aus Ihrer Observability-Lösung ziehen und gleichzeitig die Kosten kontrollieren und eine Vielzahl von Tools vermeiden. Denken Sie daran, dass Observability ein iterativer Prozess ist, kein Ziel. Richten Sie Ihre Teams und Prozesse so ein, dass Sie Ihre Beobachtbarkeit kontinuierlich iterieren und verbessern, bewährte Verfahren integrieren und Lücken schließen, während sich Ihre Architektur, Ihre Geschäftsanforderungen und Teams weiterentwickeln, wobei Sie sich an klar definierten Zielen orientieren. Wenn Sie für Misserfolge planen, eine Kultur der kontinuierlichen Verbesserung einführen und die in diesem Leitfaden erörterten Best Practices anwenden, sollten Sie in der Lage sein, die Leitlinien von Dr. Werner Vogels in die Praxis umzusetzen: „Alles schlägt ständig fehl... planen Sie also für Misserfolge und nichts wird scheitern.“

Für einen praxisnahen Workshop, der Ihnen bei der Definition Ihrer Beobachtbarkeitsstrategie hilft, wenden Sie sich an Ihren AWS Ansprechpartner, um einen Workshop zur Observabilitätsstrategie abzuhalten.

## Ressourcen

Weitere Informationen und Anleitungen zur Umsetzung einer effektiven Strategie zur Beobachtung finden Sie in den folgenden AWS Ressourcen:

- [Überwachung und Beobachtbarkeit \(Dienste zur AWS Beobachtbarkeit\)](#)
- [AWS Bewährte Verfahren zur Beobachtung](#)
- [AWS Distribution für OpenTelemetry](#)
- [Säule der operativen Exzellenz \(AWS Well-Architected Framework\)](#)
- [Erstellung von Dashboards für betriebliche Transparenz \(The Amazon Builders' Library\)](#)
- [Beobachtbarkeit \(in der DevOpsAnleitung, AWS Well-Architected Framework\)](#)

## Dokumentverlauf

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

Änderung	Beschreibung	Datum
<a href="#"><u>Erste Veröffentlichung</u></a>	—	16. Juli 2025

# AWS Glossar zu präskriptiven Leitlinien

Die folgenden Begriffe werden häufig in Strategien, Leitfäden und Mustern verwendet, die von AWS Prescriptive Guidance bereitgestellt werden. 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 verarbeitet Transaktionen von verbundenen Anwendungen, 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 im Bereich künstliche 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 in der AWS Migrationsstrategie finden Sie im [Operations Integration Guide](#). AIOps

## Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorismus, 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 Datenvielfalt 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 Einführung der Cloud (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 in der Regel 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 AWS für die 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 Störungseignissen, 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-Exzellenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantriebt, 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 — Tätigen Sie grundlegende Investitionen, 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 Bitbucket Cloud. 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. Amazon SageMaker AI bietet beispielsweise Bildverarbeitungsalgorithmen für CV.

## Drift in der Konfiguration

Bei einer Arbeitslast eine Änderung der Konfiguration gegenüber dem erwarteten Zustand. Dies kann dazu führen, dass der Workload nicht mehrrichtlinienkonform 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 Variation zwischen den Produktionsdaten und den Daten, die zum Trainieren eines ML-Modells verwendet wurden, oder eine signifikante Änderung der Eingabedaten im Laufe der Zeit. Datendrift kann die Gesamtqualität, Genauigkeit und Fairness von ML-Modellvorhersagen beeinträchtigen.

## Daten während der Übertragung

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

## Datennetz

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

## Datenminimierung

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

## Datenperimeter

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

## Vorverarbeitung der Daten

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

## Herkunft der Daten

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

## betroffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

## Data Warehouse

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

## Datenbankdefinitionssprache (DDL)

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

## Datenbankmanipulationssprache (DML)

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

## DDL

Siehe [Datenbankdefinitionssprache](#).

## Deep-Ensemble

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

## Deep Learning

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

## defense-in-depth

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

## delegierter Administrator

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

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

## Einsatz

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 Sicherheitseignissen, 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, wie z. B. unbeabsichtigte Fehlkonfigurationen oder ein Malware-Angriff.

## Disaster Recovery (DR)

Die Strategie und der Prozess, mit denen Sie Ausfallzeiten und Datenverluste aufgrund einer [Katastrophe](#) minimieren. Weitere Informationen finden Sie unter [Disaster Recovery von Workloads unter AWS: Wiederherstellung in der Cloud im AWS Well-Architected Framework](#).

## DML

Siehe Sprache zur [Datenbankmanipulation](#).

## Domainorientiertes Design

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

## DR

Siehe [Disaster Recovery](#).

## Erkennung von Driften

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

## DVSM

Siehe [Abbildung des Wertstroms in der Entwicklung](#).

# E

## EDA

Siehe [explorative Datenanalyse](#).

## EDI

Siehe [elektronischer Datenaustausch](#).

## 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.

## elektronischer Datenaustausch (EDI)

Der automatisierte Austausch von Geschäftsdokumenten zwischen Organisationen. Weitere Informationen finden Sie unter [Was ist elektronischer Datenaustausch](#).

## 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 Endpunkt mit anderen AWS-Konten oder AWS Identity and Access Management (IAM AWS PrivateLink -) Prinzipalnen erstellen und diesen

Berechtigungen gewähren. Diese Konten oder Prinzipale können sich privat mit Ihrem Endpunkt-Service 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 Sicherheitselementen 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 Fertigungsline 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.

## Eingabeaufforderung mit wenigen Klicks

Bereitstellung einer kleinen Anzahl von Beispielen für ein [LLM](#), die die Aufgabe und das gewünschte Ergebnis veranschaulichen, bevor es aufgefordert wird, eine ähnliche Aufgabe auszuführen. Bei dieser Technik handelt es sich um eine Anwendung des kontextbezogenen Lernens, bei der Modelle anhand von Beispielen (Aufnahmen) lernen, die in Eingabeaufforderungen eingebettet sind. Bei Aufgaben, die spezifische Formatierungs-, Argumentations- oder Fachkenntnisse erfordern, kann die Eingabeaufforderung mit wenigen Handgriffen effektiv sein. [Siehe auch Zero-Shot-Eingabeaufforderung](#).

## 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.

## FM

Siehe [Fundamentmodell](#).

## Fundamentmodell (FM)

Ein großes neuronales Deep-Learning-Netzwerk, das mit riesigen Datensätzen generalisierter und unbeschrifteter Daten trainiert wurde. FMs sind in der Lage, eine Vielzahl allgemeiner Aufgaben zu erfüllen, z. B. Sprache zu verstehen, Text und Bilder zu generieren und Konversationen in natürlicher Sprache zu führen. Weitere Informationen finden Sie unter [Was sind Foundation-Modelle](#).

# G

## Generative KI

Eine Untergruppe von [KI-Modellen](#), die mit großen Datenmengen trainiert wurden und mit einer einfachen Textaufforderung neue Inhalte und Artefakte wie Bilder, Videos, Text und Audio erstellen können. Weitere Informationen finden Sie unter [Was ist Generative KI](#).

## 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.

## goldenes Bild

Ein Snapshot eines Systems oder einer Software, der als Vorlage für die Bereitstellung neuer Instanzen dieses Systems oder dieser Software verwendet wird. In der Fertigung kann ein Golden Image beispielsweise zur Bereitstellung von Software auf mehreren Geräten verwendet werden und trägt so zur Verbesserung der Geschwindigkeit, Skalierbarkeit und Produktivität bei der Geräteherstellung bei.

## 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 dazu beiträgt, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Unternehmenseinheiten zu regeln (OUs). 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 CSPM, 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, eine 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.

## Daten zurückhalten

Ein Teil historischer, beschrifteter Daten, der aus einem Datensatz zurückgehalten wird, der zum Trainieren eines Modells für [maschinelles Lernen](#) verwendet wird. Sie können Holdout-Daten verwenden, um die Modellleistung zu bewerten, indem Sie die Modellvorhersagen mit den Holdout-Daten vergleichen.

## Homogene Datenbankmigration

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

## heiße Daten

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

## Hotfix

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

## Hypercare-Phase

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

I

laC

Sehen Sie [Infrastruktur als Code](#).

Identitätsbasierte Richtlinie

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

Leerlaufanwendung

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

IIoT

Siehe [Industrielles Internet der Dinge](#).

unveränderliche Infrastruktur

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

Eingehende (ingress) VPC

In einer Architektur AWS mit mehreren Konten ist dies eine VPC, die Netzwerkverbindungen von außerhalb einer Anwendung akzeptiert, überprüft und weiterleitet. Die [AWS Security Reference Architecture](#) empfiehlt, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr und Inspektion einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne 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 (T) IIo

Einsatz von mit dem Internet verbundenen Sensoren und Geräten in Industriesektoren wie Fertigung, Energie, Automobilindustrie, Gesundheitswesen, Biowissenschaften und Landwirtschaft. Weitere 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 demselben oder unterschiedlichen AWS-Regionen), dem Internet und lokalen Netzwerken verwaltet. In der [AWS Security Reference Architecture](#) wird empfohlen, Ihr Netzwerkonto mit eingehendem und ausgehendem Datenverkehr sowie Inspektionen einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne 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 des [Modells für maschinelles Lernen](#) 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.

## IT service management (ITSM, IT-Servicemanagement)

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

## BIS

Weitere Informationen finden Sie in der [IT-Informationsbibliothek](#).

## ITSM

Siehe [IT-Servicemanagement](#).

## L

## Labelbasierte Zugangskontrolle (LBAC)

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

## Landing Zone

Eine landing zone ist eine gut strukturierte AWS Umgebung mit mehreren Konten, die skalierbar und sicher ist. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und Infrastrukturumgebung starten

und bereitstellen können. Weitere Informationen zu Landing Zones finden Sie unter [Einrichtung einer sicheren und skalierbaren AWS -Umgebung mit mehreren Konten..](#)

großes Sprachmodell (LLM)

Ein [Deep-Learning-KI-Modell](#), das anhand einer riesigen Datenmenge vorab trainiert wurde. Ein LLM kann mehrere Aufgaben ausführen, z. B. Fragen beantworten, Dokumente zusammenfassen, Text in andere Sprachen übersetzen und Sätze vervollständigen. [Weitere Informationen finden Sie unter Was sind. LLMs](#)

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.](#)

LLM

Siehe [großes Sprachmodell.](#)

Niedrigere Umgebungen

Siehe [Umgebung.](#)

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 Mitglied einer Organisation sein.

## MES

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 Dienst, der über genau definierte Kanäle kommuniziert APIs 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 mithilfe von Lightweight über eine klar definierte Schnittstelle. 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

[Siehe Origin Access Control.](#)

## OAII

[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.

## Ol

[Siehe Betriebsintegration.](#)

## OLA

[Siehe Vereinbarung auf operativer Ebene.](#)

## Online-Migration

Eine Migrationsmethode, bei der der Quell-Workload auf das Zielsystem kopiert wird, ohne offline genommen zu werden. Anwendungen, die mit dem Workload verbunden sind, können während

der Migration weiterhin funktionieren. Diese Methode beinhaltet keine bis minimale Ausfallzeit und wird in der Regel für kritische Produktionsworkloads verwendet.

## OPC-UA

Siehe [Open Process Communications — Unified Architecture](#).

### Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

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

## Vereinbarung auf Betriebsebene (OLA)

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

## Überprüfung der Betriebsbereitschaft (ORR)

Eine Checkliste mit Fragen und zugehörigen bewährten Methoden, die Ihnen helfen, Vorfälle und mögliche Ausfälle zu verstehen, zu bewerten, zu verhindern oder deren Umfang zu reduzieren.

Weitere Informationen finden Sie unter [Operational Readiness Reviews \(ORR\)](#) im AWS Well-Architected Framework.

## Betriebstechnologie (OT)

Hardware- und Softwaresysteme, die mit der physischen Umgebung zusammenarbeiten, um industrielle Abläufe, Ausrüstung und Infrastruktur zu steuern. In der Fertigung ist die Integration von OT- und Informationstechnologie (IT) -Systemen ein zentraler Schwerpunkt der [Industrie 4.0-Transformationen](#).

## Betriebsintegration (OI)

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

## Organisationspfad

Ein Pfad, der von erstellt wird und in AWS CloudTrail dem alle Ereignisse für alle AWS-Konten in einer Organisation protokolliert werden. AWS Organizations Diese Spur wird in jedem AWS-Konto, der Teil der Organisation ist, erstellt und verfolgt die Aktivität in jedem Konto. Weitere Informationen finden Sie in der CloudTrail Dokumentation unter [Einen Trail für eine Organisation erstellen](#).

## 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.

## Ursprungszugriffssidentität (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.

## ORR

Weitere Informationen finden Sie unter [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 Security Reference Architecture](#) empfiehlt die Einrichtung Ihres Netzwerkkontos mit eingehendem und ausgehendem Datenverkehr sowie Inspektion, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

# P

## Berechtigungsgrenze

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

## persönlich identifizierbare Informationen (PII)

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

## Personenbezogene Daten

Siehe [persönlich identifizierbare Informationen](#).

## Playbook

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

## PLC

Siehe [programmierbare Logiksteuerung](#).

## PLM

Siehe [Produktlebenszyklusmanagement](#).

## policy

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

## Polyglotte Beharrlichkeit

Unabhängige Auswahl der Datenspeichertechnologie eines Microservices auf der Grundlage von Datenzugriffsmustern und anderen Anforderungen. Wenn Ihre Microservices über dieselbe

Datenspeichertechnologie verfügen, kann dies zu Implementierungsproblemen oder zu Leistungseinbußen führen. Microservices lassen sich leichter implementieren und erzielen eine bessere Leistung und Skalierbarkeit, wenn sie den Datenspeicher verwenden, der ihren Anforderungen am besten entspricht. Weitere Informationen finden Sie unter [Datenpersistenz in Microservices aktivieren](#).

## Portfoliobewertung

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

Eine Abfragebedingung, die true oder zurückgibtfalse, 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 Verteidigungsline 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 von Natur aus

Ein systemtechnischer Ansatz, der den Datenschutz während des gesamten Entwicklungsprozesses berücksichtigt.

## Privat gehostete Zonen

Ein Container, der Informationen darüber enthält, wie Amazon Route 53 auf DNS-Abfragen für eine Domain und deren Subdomains innerhalb einer oder mehrerer VPCs Domains antworten

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.

## schnelle Verkettung

Verwendung der Ausgabe einer [LLM-Eingabeaufforderung](#) als Eingabe für die nächste Aufforderung, um bessere Antworten zu generieren. Diese Technik wird verwendet, um eine komplexe Aufgabe in Unteraufgaben zu unterteilen oder um eine vorläufige Antwort iterativ zu verfeinern oder zu erweitern. Sie trägt dazu bei, die Genauigkeit und Relevanz der Antworten eines Modells zu verbessern und ermöglicht detailliertere, personalisierte Ergebnisse.

## 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.

## publish/subscribe (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 \(RACI\)](#).

### LAPPEN

Siehe [Erweiterte Generierung beim Abrufen](#).

### 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. Damit wird festgelegt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Serviceunterbrechung gilt.

## Wiederherstellungszeitziel (RTO)

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

Refaktorierung

Siehe [7 Rs.](#)

## Region

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

Regression

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

rehosten

Siehe [7 Rs.](#)

## Veröffentlichung

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

umziehen

Siehe [7 Rs.](#)

neue Plattform

Siehe [7 Rs.](#)

Rückkauf

Siehe [7 Rs.](#)

Ausfallsicherheit

Die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich von ihnen zu erholen.

[Hochverfügbarkeit](#) und [Notfallwiederherstellung](#) sind häufig Überlegungen bei der Planung der Ausfallsicherheit in der AWS Cloud. Weitere Informationen finden Sie unter [AWS Cloud Resilienz](#).

Ressourcenbasierte Richtlinie

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

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

Eine Matrix, die die Rollen und Verantwortlichkeiten für alle Parteien definiert, die an Migrationsaktivitäten und Cloud-Vorgängen beteiligt sind. Der Matrixname leitet sich von den in der Matrix definierten Zuständigkeitsarten 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.](#)

## Retrieval Augmented Generation (RAG)

Eine [generative KI-Technologie](#), bei der ein [LLM](#) auf eine maßgebliche Datenquelle verweist, die sich außerhalb seiner Trainingsdatenquellen befindet, bevor eine Antwort generiert wird. Ein RAG-Modell könnte beispielsweise eine semantische Suche in der Wissensdatenbank oder in benutzerdefinierten Daten einer Organisation durchführen. Weitere Informationen finden Sie unter [Was ist RAG](#).

## 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 für die Erholungszeit](#).

## 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-Managementkonsole 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 Benutzeranmeldedaten, 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.

## Sicherheit durch Design

Ein systemtechnischer Ansatz, der die Sicherheit während des gesamten Entwicklungsprozesses berücksichtigt.

## Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungskreators, 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 Features 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 Sicherheitereignissen (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 Sicherheitereignis 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 EC2 Amazon-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 Steuerung der Berechtigungen für alle Konten in einer Organisation ermöglicht AWS Organizations. SCPs Definieren Sie Leitplanken oder legen Sie Grenzwerte für Aktionen fest, die ein Administrator an Benutzer oder Rollen delegieren kann. Sie können sie SCPs als Zulassungs- oder Ablehnungslisten verwenden, um festzulegen, welche Dienste oder Aktionen zulässig oder verboten sind. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter [Richtlinien zur Dienststeuerung](#).

### Service-Endpunkt

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

### Service Level Agreement (SLA)

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

### Service-Level-Indikator (SLI)

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

### Service-Level-Ziel (SLO)

Eine Zielkennzahl, die den Zustand eines Dienstes darstellt, gemessen anhand eines [Service-Level-Indikators](#).

### Modell der geteilten Verantwortung

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

## SIEM

Siehe [Sicherheitsinformations- und Event-Management-System](#).

## Single Point of Failure (SPOF)

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

## SLA

Siehe [Service Level Agreement](#).

## SLI

Siehe [Service-Level-Indikator](#).

## ALSO

Siehe [Service-Level-Ziel](#).

## split-and-seed Modell

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

## SPOTTEN

Siehe [Single Point of Failure](#).

## Sternschema

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

## Strangler-Fig-Muster

Ein Ansatz zur Modernisierung monolithischer Systeme, bei dem die Systemfunktionen schrittweise umgeschrieben und ersetzt werden, bis das Legacy-System außer Betrieb

genommen werden kann. Dieses Muster verwendet die Analogie einer Feigenrebe, die zu einem etablierten Baum heranwächst und schließlich ihren Wirt überwindet und ersetzt. Das Muster wurde [eingeführt von Martin Fowler](#) als Möglichkeit, Risiken beim Umschreiben monolithischer Systeme zu managen. Ein Beispiel für die Anwendung dieses Musters finden Sie unter [Schrittweises Modernisieren älterer Microsoft ASP.NET \(ASMX\)-Webservices mithilfe von Containern und Amazon API Gateway](#).

## Subnetz

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

## Aufsichtskontrolle und Datenerfassung (SCADA)

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

## Symmetrische Verschlüsselung

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

## synthetisches Testen

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

## Systemaufforderung

Eine Technik, mit der einem [LLM](#) Kontext, Anweisungen oder Richtlinien zur Verfügung gestellt werden, um sein Verhalten zu steuern. Systemaufforderungen helfen dabei, den Kontext festzulegen und Regeln für Interaktionen mit Benutzern festzulegen.

## 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 Netzwerk-Transit-Hub, über den Sie Ihre Netzwerke VPCs und Ihre lokalen Netzwerke miteinander verbinden können. Weitere Informationen finden Sie in der Dokumentation unter [Was ist ein Transit-Gateway](#). AWS 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, die es Ihnen ermöglicht, den Verkehr mithilfe privater IP-Adressen weiterzuleiten. Weitere Informationen finden Sie unter [Was ist VPC-Peering?](#) in der Amazon-VPC-Dokumentation.

## Schwachstelle

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

## 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

Siehe [AWS Workload-Qualifizierungsrahmen.](#)

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 wird als [unveränderlich](#) angesehen.

## 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.

Bedrohungskräfte können diese Art von Sicherheitslücke nutzen, um das System anzugreifen. Entwickler werden aufgrund des Angriffs häufig auf die Sicherheitsanfälligkeit aufmerksam.

### Eingabeaufforderung ohne Zwischenfälle

Bereitstellung von Anweisungen für die Ausführung einer Aufgabe an einen [LLM](#), jedoch ohne Beispiele (Schnappschüsse), die ihm als Orientierungshilfe dienen könnten. Der LLM muss sein

vortrainiertes Wissen einsetzen, um die Aufgabe zu bewältigen. Die Effektivität von Zero-Shot Prompting hängt von der Komplexität der Aufgabe und der Qualität der Aufforderung ab. [Siehe auch Few-Shot-Eingabeaufforderungen.](#)

## 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.