



Integration von Microservices mithilfe serverloser Dienste AWS

AWS Präskriptive Leitlinien



AWS Präskriptive Leitlinien: Integration von Microservices mithilfe serverloser Dienste AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irreführt 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
Sicherheit	2
Kommunikationsmuster	3
Synchrone Kommunikation	3
Asynchrone Kommunikation	5
Abfeuern und vergessen	5
Prüfung der Reklamation	6
Callback	7
Bidirektionale Kommunikation	8
Möglichkeiten der Koordination	11
Orchestrierung	11
Beispiel: Step Functions	11
Beispiel: Amazon MWAA	14
Hauptunterschiede zwischen Step Functions und Amazon MWAA	16
Choreographie	16
Wählen Sie Ihren Koordinationsansatz	17
Verwaltung APIs	18
Amazon API Gateway	18
Authentifizierung und Autorisierung	18
API-Schlüssel und Ratenbegrenzungen	19
Öffentlich und privat APIs	19
Wann sollte API Gateway verwendet werden	20
Messaging	21
Amazon SQS	21
Abrufen	22
Empfehlungen	22
Amazon SNS	23
Empfehlungen	24
Amazon EventBridge	24
Empfehlungen	25
AWS AppSync Ereignisse und API Gateway	26
Empfehlungen	26

Häufig gestellte Fragen	28
Wie kann ich verschiedene Integrationsmuster kombinieren?	28
Was ist der Hauptvorteil der Verwendung einer Microservices-Architektur?	28
Wie kann ich die Fehlerbehandlung implementieren?	28
Was sind die Vorteile des Musters zur Schadensüberprüfung?	28
Was sind die Vorteile des Callback-Musters?	29
Kann ich bidirektionale Kommunikation implementieren?	29
Wie kann ich die Verwendung von Lambda-Funktionen optimieren?	29
Was sind die Hauptunterschiede zwischen der Verwendung von Amazon SNS und EventBridge?	29
Ressourcen	31
AWS-Service Dokumentation	31
Zusätzliche Lektüre	31
Dokumentverlauf	32
Glossar	33
#	33
A	34
B	37
C	39
D	43
E	47
F	49
G	51
H	53
I	54
L	57
M	58
O	62
P	65
Q	68
R	69
S	72
T	76
U	78
V	78
W	79

Z	80
.....	lxxxi

Integration von Microservices mithilfe AWS serverloser Dienste

Tabby Ward, Abhishek Agawane und Matt Kahn, Amazon Web Services

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

Ein wichtiger Aspekt bei der Modernisierung der Software Ihres Unternehmens ist die Auswahl der richtigen Architekturmuster, die Flexibilität und Reaktionsfähigkeit auf sich ändernde Geschäftsanforderungen ermöglichen. In einigen Anwendungen ist eine monolithische Architektur eine gängige Wahl. Für viele Unternehmen kann die [Umgestaltung von Monolithen in Microservices](#) jedoch eine effektive Modernisierungsstrategie sein, wenn der Anwendungsfall mit den Vorteilen von Microservices übereinstimmt.

Microservices und Monolithen schließen sich nicht gegenseitig aus — viele erfolgreiche Unternehmen nutzen beide Muster zusammen, wobei modulare Monolithen für einige Bereiche und Microservices für andere Bereiche zuständig sind.

Wenn Microservices Teil Ihrer Architektur sind, werden möglicherweise mehrere Dienste aufgerufen, um Daten für eine Geschäftstransaktion abzurufen. Die Implementierung dieser Integrationen erfordert ein sorgfältiges Design, um potenziellen Herausforderungen wie Datenkonsistenz, Latenz und betrieblicher Komplexität zu begegnen. Wenn Microservices richtig integriert sind, können sie Vorteile wie unabhängige Skalierung, verbesserte Entwicklungsgeschwindigkeit und potenzielle Kostenoptimierungen bieten.

Der Leitfaden ist Teil einer Inhaltsreihe, die den von empfohlenen Ansatz zur Anwendungsmodernisierung behandelt. AWS Die Serie umfasst auch:

- [Strategie zur Modernisierung von Anwendungen in der AWS Cloud](#)
- [Stufenweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud](#)
- [Bewertung der Modernisierungsbereitschaft von Anwendungen in der AWS Cloud](#)
- [Zerlegung von Monolithen in Microservices](#)

Zielgruppe

Dieser Leitfaden richtet sich an Anwendungsinhaber, Geschäftsinhaber, Architekten, technische Leiter und Projektmanager, die festgestellt haben, dass Microservices für ihren spezifischen

Anwendungsfall geeignet sind. In diesem Leitfaden werden verschiedene Muster für synchrone und asynchrone Kommunikation zwischen Microservices vorgestellt, bei denen serverlose Dienste AWS-Services wie Amazon API Gateway verwendet werden, um AWS Lambda Autonomie und Skalierbarkeit zu gewährleisten.

Ziele

Mithilfe dieses Leitfadens zur Integration Ihrer neuen Microservices können Sie die Architektur Ihres Unternehmens effizient in eine Microservices-Architektur umwandeln. Dies trägt durch hohe Skalierbarkeit, verbesserte Ausfallsicherheit, kontinuierliche Bereitstellung und Ausfallsolierung zu einer schnellen Anpassung an schwankende Geschäftsanforderungen bei. Eine Microservices-Architektur beschleunigt auch Innovationen, da jeder Microservice einzeln bereitgestellt und getestet werden kann.

Eine Microservices-Architektur kann auch dazu beitragen, die Markteinführungszeit Ihrer Produkte oder Dienstleistungen zu verkürzen, da jeder Microservice über eine unabhängige Codebasis verfügt, die es einfacher und schneller macht, neue Funktionen hinzuzufügen und zu iterieren.

Sicherheit

Sie müssen Ihre Microservices ordnungsgemäß sichern, um die Integrität Ihrer Dienste und Daten zu schützen, aber sicherstellen, dass sich die Sicherheit nicht negativ auf die Leistung Ihrer Anwendung auswirkt.

In einer Microservices-Umgebung müssen Sie berücksichtigen, wie jeder Service Anfragen authentifiziert und autorisiert, die er von einem externen Client oder einem anderen Microservice erhält. Überlegen Sie auch, wie die einzelnen Dienste sicher auf andere zugreifen können. AWS-Services

Der Zugriff auf AWS-Services sollte über eng begrenzte Rollen [AWS Identity and Access Management](#) (IAM) gewährt werden. Angenommen, eine IAM-Rolle stellt dem Microservice kurzfristige IAM-Anmeldeinformationen in Form eines Zugriffsschlüssels, eines Zugriffsgeheimnisses und eines Sitzungstokens zur Verfügung. Diese werden von den verschiedenen Software Development Kits (SDKs) verwendet, um Anfragen mithilfe AWS-Services von [AWS Signature Version 4](#) (Sigv4) zu signieren.

Kommunikationsmuster

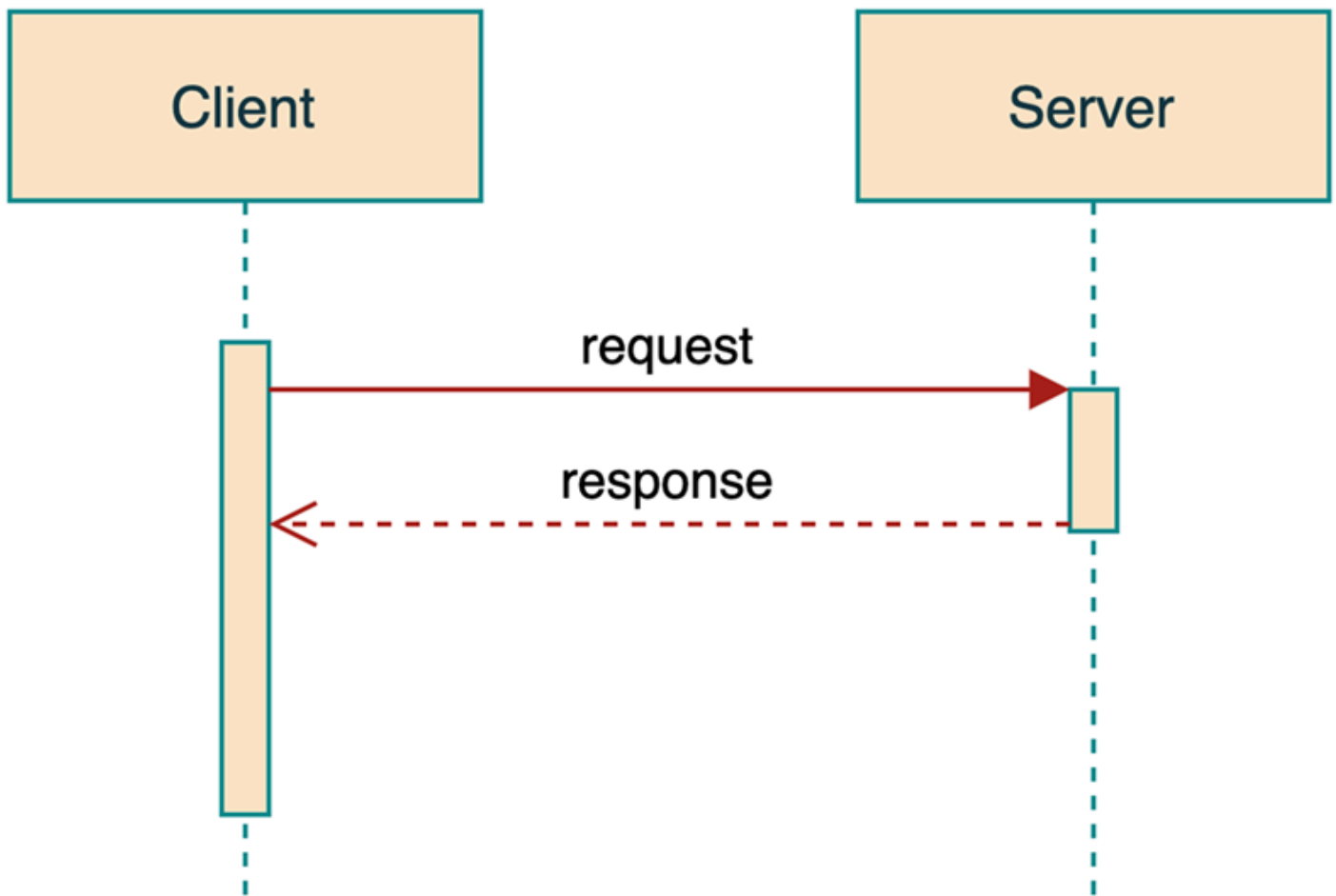
In einer Microservices-Architektur erfolgt die Kommunikation in zwei Hauptmustern: synchron und asynchron. Bei der synchronen Kommunikation wartet der Aufrufer auf eine Antwort, bevor er fortfährt, ähnlich wie bei einem HTTP-REST-API-Aufruf in Echtzeit. Die asynchrone Kommunikation folgt einem nachrichtenbasierten Muster, bei dem der Aufrufer die Verarbeitung fortsetzt, ohne auf eine Antwort zu warten, z. B. bei der Verwendung von Nachrichtenwarteschlangen. In den folgenden Abschnitten werden die Implementierung, die Vorteile und die Anwendungsfälle der einzelnen Muster im Detail untersucht.

Themen

- [Synchrone Kommunikation](#)
- [Asynchrone Kommunikation](#)

Synchrone Kommunikation

Bei der synchronen Kommunikation initiiert ein Client eine Anfrage an einen Dienst, wie das folgende Diagramm zeigt. Beispiele hierfür sind eine Anforderung zum Abrufen von Informationen, z. B. eine HTTP GET Anforderung, oder eine Anforderung zur Änderung von Daten, z. B. eine Anforderung. HTTP PUT In beiden Fällen wartet der Client auf die Antwort des Servers, bevor er fortfährt. Synchrone Aufrufe sind den meisten Entwicklern vertraut, sie lassen sich leicht implementieren und Fehler beheben und sind in vielen Fällen der allgemein anerkannte Kommunikationsstandard.



Zu den Vorteilen der synchronen Kommunikation gehören:

- Vorhersehbare Flusskontrolle — Deterministische Ausführung und klare Anfrage-Antwort-Zyklen, die im Vergleich zur asynchronen Kommunikation leichter zu verstehen sind.
- Starke Konsistenz — Sofortige Bestätigung von Datenänderungen und Statusaktualisierungen.
- Einfache Fehlerbehandlung — Direkte Weitergabe von Fehlern und Ausnahmen.
- Einfaches Debuggen — Unkomplizierte Nachverfolgung und Überwachung von Anfragen.
- Protokollunterstützung — Etablierte Protokolle wie HTTP und REST, die eine einfache Implementierung ermöglichen.

Synchrone Kommunikation hat einige Nachteile:

- Enge Kopplung — Direkte Abhängigkeit der Dienste von der Verfügbarkeit des jeweils anderen.
- Auswirkungen auf das Netzwerk — Erhöhte Netzwerklast aufgrund ständig offener Verbindungen.

- Ressourcenauslastung — Höherer Speicherverbrauch durch Beibehaltung des Verbindungsstatus.
- Kaskadierende Ausfälle — Probleme in einem Dienst können sich schnell im System ausbreiten.

Asynchrone Kommunikation

Umgekehrt sendet der Client bei asynchroner Kommunikation eine Anfrage an einen Dienst, erhält aber keine sofortige Antwort. In diesem Fall erhält der Client normalerweise nur eine Bestätigung, dass die Anfrage akzeptiert wurde.

Zu den Vorteilen der asynchronen Kommunikation gehören:

- Unterstützung ereignisgesteuerter Architekturen — Eignet sich hervorragend für die Ereignisbeschaffung und CQRS-Muster (Command Query Responsibility Segregation).
- Besseres Ressourcenmanagement — Fähigkeit der Dienste, Anfragen auf der Grundlage ihrer Kapazität zu bearbeiten.
- Verbesserte Fehlerisolierung — Entkopplung von Diensten, wodurch Kaskadenausfälle verhindert werden.
- Umgang mit Spitzenlasten — Besserer Umgang mit Datenverkehrsspitzen durch Message Queuing.

Zu den Nachteilen gehört die Komplexität. Beispiel:

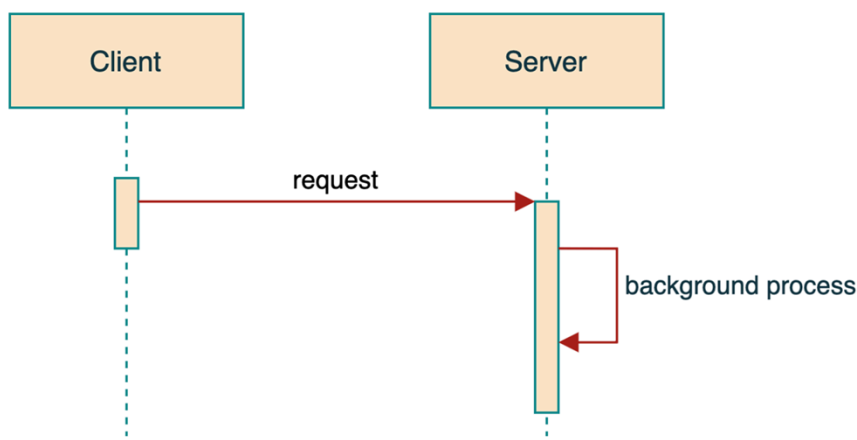
- Wenn der Client das Ergebnis des asynchronen Vorgangs benötigt, erfordert die Implementierung eines Mechanismus zum Abrufen oder Empfangen dieses Ergebnisses mehr Aufwand.
- Es kann schwieriger sein, Fehler bei asynchronen Vorgängen zu beheben, da die Fehlerbehebung die Untersuchung von Protokollen mehrerer Systeme erfordert.
- Es kann schwieriger sein, asynchrone Operationen zu testen, da das Testen die Koordination zwischen mehreren Systemen und Diensten erfordert.

Zu den Ansätzen für asynchrone Kommunikation gehören Fire and Forget, Claim Check, Callback und bidirektionale Kommunikation.

Abfeuern und vergessen

Beim Fire-and-Forget-Muster sendet ein Client eine Anfrage an den Server und erhält synchron eine Bestätigung, dass der Server die Nachricht erhalten hat und verarbeiten wird. Die eigentliche

Verarbeitung hat jedoch noch nicht stattgefunden, und der Client hat keine Vorstellung davon, wann oder wie sie durchgeführt wird. Das folgende Diagramm veranschaulicht dieses Muster.



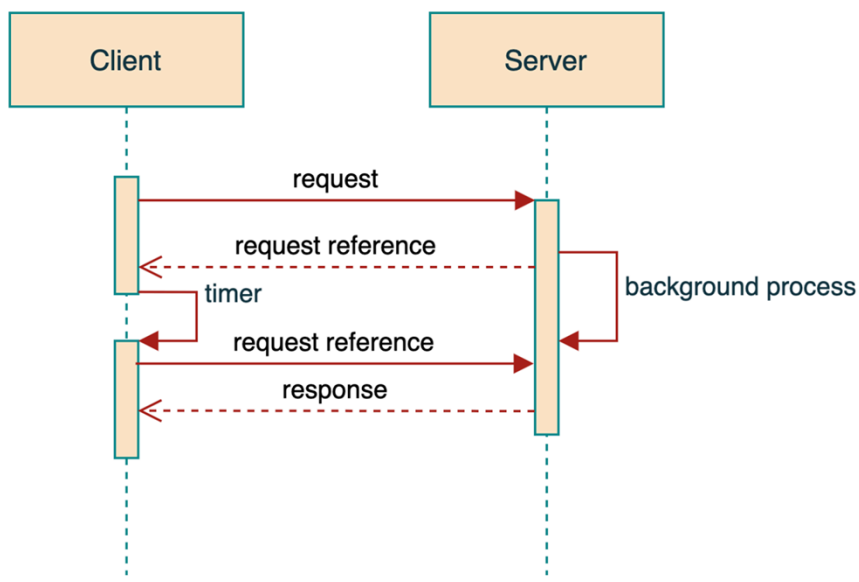
In diesem Fall sollte der Dienst die Bestätigung erst senden, wenn das Objekt dauerhaft persistent ist. Diese Persistenz könnte als Datenbank-Schreibvorgang oder durch das Einfügen eines Elements in eine Warteschlange implementiert werden.

Weitere Überlegungen:

- Implementieren Sie Idempotenz, um doppelte Nachrichten zu verarbeiten. Das heißt, jede Nachricht sollte nur einmal verarbeitet werden.
- Denken Sie bei [fehlgeschlagenen Verarbeitungsvorgängen an Warteschlangen für unzustellbare Briefe](#).
- Überwachen Sie die Erfolgsquoten der Nachrichtenverarbeitung.

Prüfung der Reklamation

Wenn ein Kunde das Ergebnis eines Serviceanrufs benötigt, können Sie den Service so einrichten, dass er bei Eingang einer Anfrage einen Schadensnachweis ausstellt. Das folgende Diagramm veranschaulicht dieses Muster. Die Anspruchsprüfung wird als Kennung implementiert, die der Dienst in seiner Bestätigung zurückgibt. Der Client kann diese Kennung später verwenden, um den Status der Anfrage zu überprüfen und das Ergebnis abzurufen, wenn die Anfrage abgeschlossen ist.



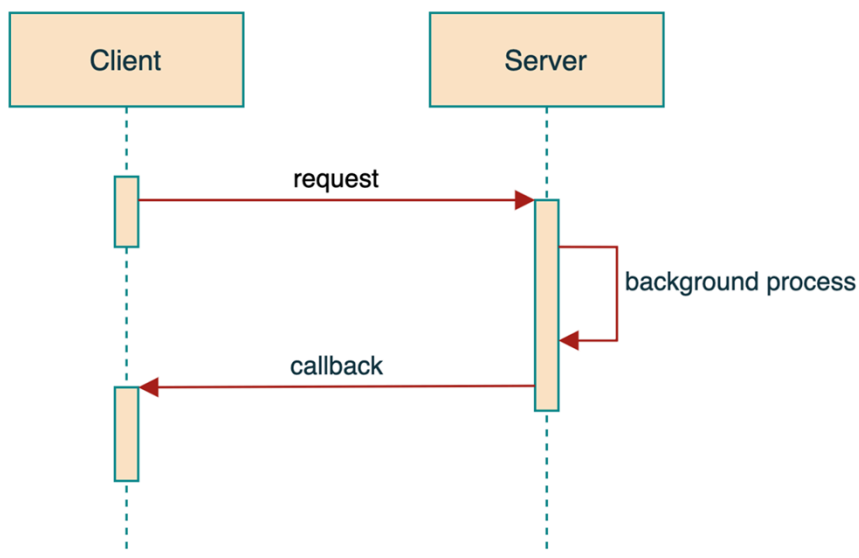
Clients müssen einen Mechanismus zur Abfrage von Ergebnissen implementieren. Dies kann automatisiert (z. B. kann eine Überprüfung alle n Minuten durchgeführt werden) oder manuell implementiert werden, wobei die Überprüfung als Reaktion auf ein anderes Ereignis oder eine Benutzeraktion durchgeführt wird. Dienste, die das Muster der Anspruchsprüfung implementieren, sollten ausdrücklich angeben, wie lange eine Anspruchsprüfung gültig ist.

Bewährte Verfahren:

- Implementieren Sie exponentielles Backoff für Abfragen.
- Legen Sie eine angemessene Gültigkeitsdauer (TTL) für Forderungsprüfungen fest.
- Stellen Sie Status-Endpunkte für die Fortschrittsverfolgung bereit.

Callback

Im Rückrufmuster sendet ein Kunde eine Anfrage an einen Service und gibt dem Service einen Standort an, an den er sich wenden kann, wenn die Bearbeitung abgeschlossen ist. Der Client wartet nicht auf ein Ergebnis und die Verarbeitung wird fortgesetzt. Der Service ist dafür verantwortlich, den Standort zu kontaktieren, wenn die Bearbeitung abgeschlossen ist, und das Ergebnis bereitzustellen. Übliche Speicherorte für Antworten sind REST APIs oder Warteschlangen. Das folgende Diagramm veranschaulicht das Rückrufmuster.

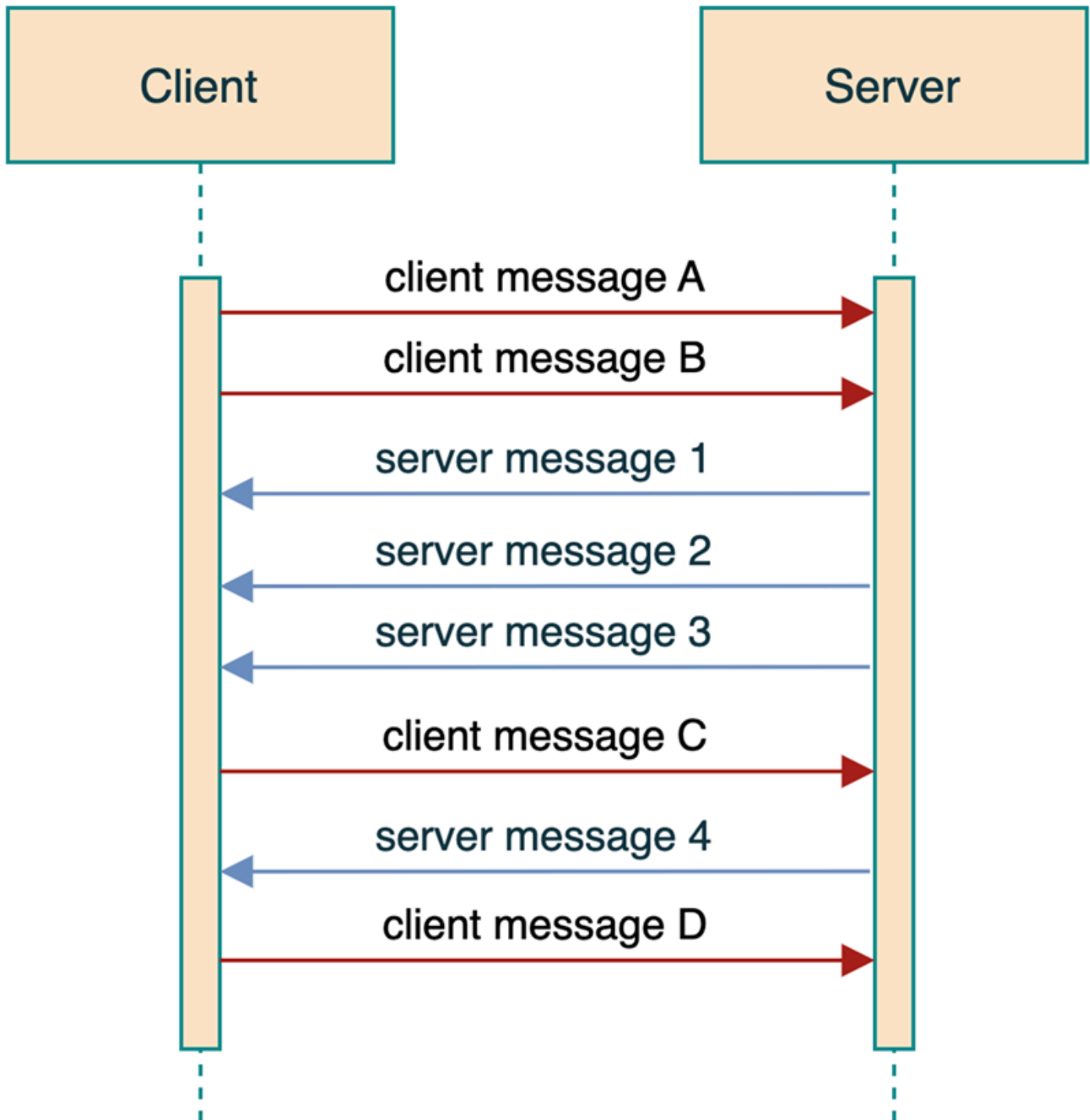


Implementierung:

- Implementieren Sie Wiederholungsmechanismen für fehlgeschlagene Rückrufe.
- Sichern Sie den Standort für den Rückruf wie bei anderen Diensten.
- Behandeln Sie Rückruf-Timeouts.

Bidirektionale Kommunikation

Um die bidirektionale Kommunikation zu implementieren, müssen Sie eine statusbehaftete Verbindung zwischen einem Client und einem Dienst herstellen, sodass sowohl der Client als auch der Dienst Nachrichten senden und verarbeiten können. Dies wird im folgenden Diagramm veranschaulicht. Obwohl die Kommunikation asynchron ist, muss der Dienst in der Lage sein, eine offene Verbindung für jeden Client zu unterstützen.



Überlegungen zur Implementierung:

- Nachrichtenreihenfolge
- Sequenznummern

- Partitionsstrategien
- Nachrichtenreihenfolge
- Statusverwaltung
 - Muster bei der Beschaffung von Veranstaltungen
 - Aussöhnung der Staaten
 - Konsistenzmodelle
- Fehlerbehandlung
 - [Warteschlangen für tote Briefe](#)
 - Wiederholungsrichtlinien
 - [Leistungsschalter](#)
 - Ausweichstrategien
- Überwachung und Beobachtbarkeit
 - Korrelation IDs
 - Nachverfolgung von Nachrichten
 - Leistungsmetriken
 - Indikatoren für den Systemzustand

Möglichkeiten der Koordination

Synchrone und asynchrone Kommunikation eignen sich gut für einen Client, der einen einzelnen Dienst oder eine kleine Anzahl von Diensten aufruft. In einer realen Umgebung kann diese Kommunikation jedoch schnell kompliziert und schwer skalierbar werden. Für die Ausführung einer Arbeitseinheit sind möglicherweise mehrere Microservices erforderlich, die wechselseitige Abhängigkeiten aufweisen können. Oft werden diese Interaktionen als Workflow modelliert. Für die Gestaltung dieser Workflows gibt es zwei Ansätze: Orchestrierung und Choreographie.

Themen

- [Orchestrierung](#)
- [Choreographie](#)
- [Wählen Sie Ihren Koordinationsansatz](#)

Orchestrierung

Bei diesem Ansatz ist ein einziger Orchestrator dafür verantwortlich, jeden Microservice aufzurufen, zu bestimmen, ob Aufrufe nacheinander oder parallel ausgeführt werden sollen, die einzelnen Serviceantworten zu manipulieren und das Endergebnis zu kompilieren. Ein Orchestrator kann synchrone und asynchrone Aufrufe kombinieren.

[AWS Step Functions](#) und [Amazon Managed Workflows for Apache Airflow \(Amazon MWAA\)](#) sind eine hervorragende Wahl für Workflow-Orchestratoren.

Orchestrierung ist eine gute Wahl, wenn Ihr Prozess logische Verzweigungen enthält und Sie einen zentralen Ort benötigen, um diese Logik zu kapseln. Sie ist auch nützlich, wenn Sie das asynchrone Muster für die Prüfung von Ansprüchen implementieren möchten. Beispielsweise können Standard-Workflows in Step Functions einen Workflow pausieren und auf einen Rückruf von einem anderen Service warten. Die Verwendung eines Orchestrators verbessert auch die Überwachung und Beobachtbarkeit eines Prozesses.

Beispiel: Step Functions

Sie können Step Functions verwenden, um mehrere Lambda-Funktionen und andere AWS-Services zu koordinieren und komplexe Workflows für die Microservice-Integration zu erstellen. Diese Option

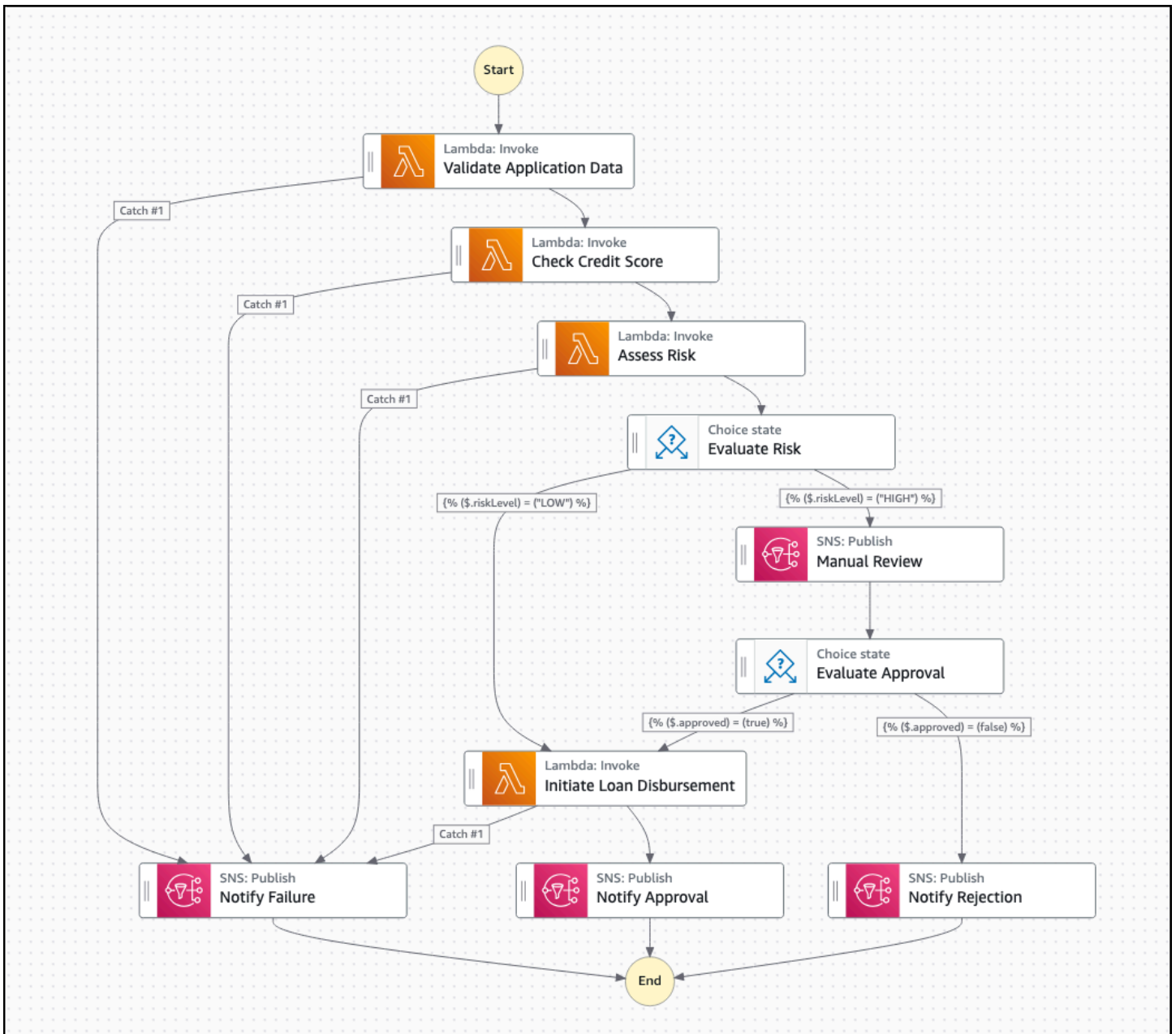
ist besonders nützlich für lang andauernde, mehrstufige Prozesse, an denen mehrere Microservices beteiligt sind.

Sie sollten die Verwendung von Step Functions in Betracht ziehen, wenn:

- Ihre Microservice-Integration umfasst komplexe, mehrstufige Prozesse.
- Sie müssen den Status bei lang andauernden Vorgängen aufrechterhalten.
- Sie möchten die Fehlerbehandlungs- und Wiederholungslogik auf Workflow-Ebene implementieren.
- Sie müssen sowohl synchrone als auch asynchrone Operationen koordinieren.

Step Functions bietet einen visuellen Editor für die Gestaltung komplexer Workflows, der die Erstellung und Verwaltung von Zustandsmaschinen vereinfacht. Er bietet integrierte Mechanismen zur Fehlerbehandlung, einschließlich Wiederholungslogik und Fehlerstatusverwaltung, die die Zuverlässigkeit und Robustheit Ihrer Anwendungen erhöhen. Standardworkflows unterstützen Prozesse mit langer Laufzeit von bis zu einem Jahr, was sich für Workflows über längere Zeiträume eignet. Diese Option trennt die Orchestrierungslogik vom Anwendungscode und reduziert so die Codekomplexität erheblich. Das bedeutet, dass sich Entwickler auf die Kerngeschäftslogik konzentrieren können, während Step Functions die Flusskontrolle und Koordination der verteilten Komponenten übernimmt.

Stellen Sie sich zum Beispiel einen Kreditgenehmigungsprozess in einer Finanzdienstleistungsanwendung vor, der in der folgenden Abbildung veranschaulicht wird. Der Prozess beginnt, wenn ein Kreditantrag eingereicht wird.



In der Zustandsmaschine, die im vorherigen Diagramm dargestellt ist, orchestriert Step Functions die folgenden Schritte:

- Anwendungsdaten validieren (Lambda-Funktion)
- Kreditwürdigkeit überprüfen (Lambda-Funktion, die eine externe API aufruft)
- Risiko einschätzen (Lambda-Funktion)
- Wenn das Risiko hoch ist, leiten Sie es zur manuellen Überprüfung weiter (Aufgabe zur Genehmigung durch einen Mitarbeiter)

- Falls genehmigt, initiieren Sie die Kreditauszahlung (Lambda-Funktion)
- Benachrichtigung an den Antragsteller senden (Amazon SNS)

Sie können diesen Ansatz verwenden, um einen komplexen, potenziell lang andauernden Prozess zuverlässig zu verwalten. Er bietet eine integrierte Fehlerbehandlung und die Möglichkeit, sowohl automatisierte als auch manuelle Schritte einzubeziehen.

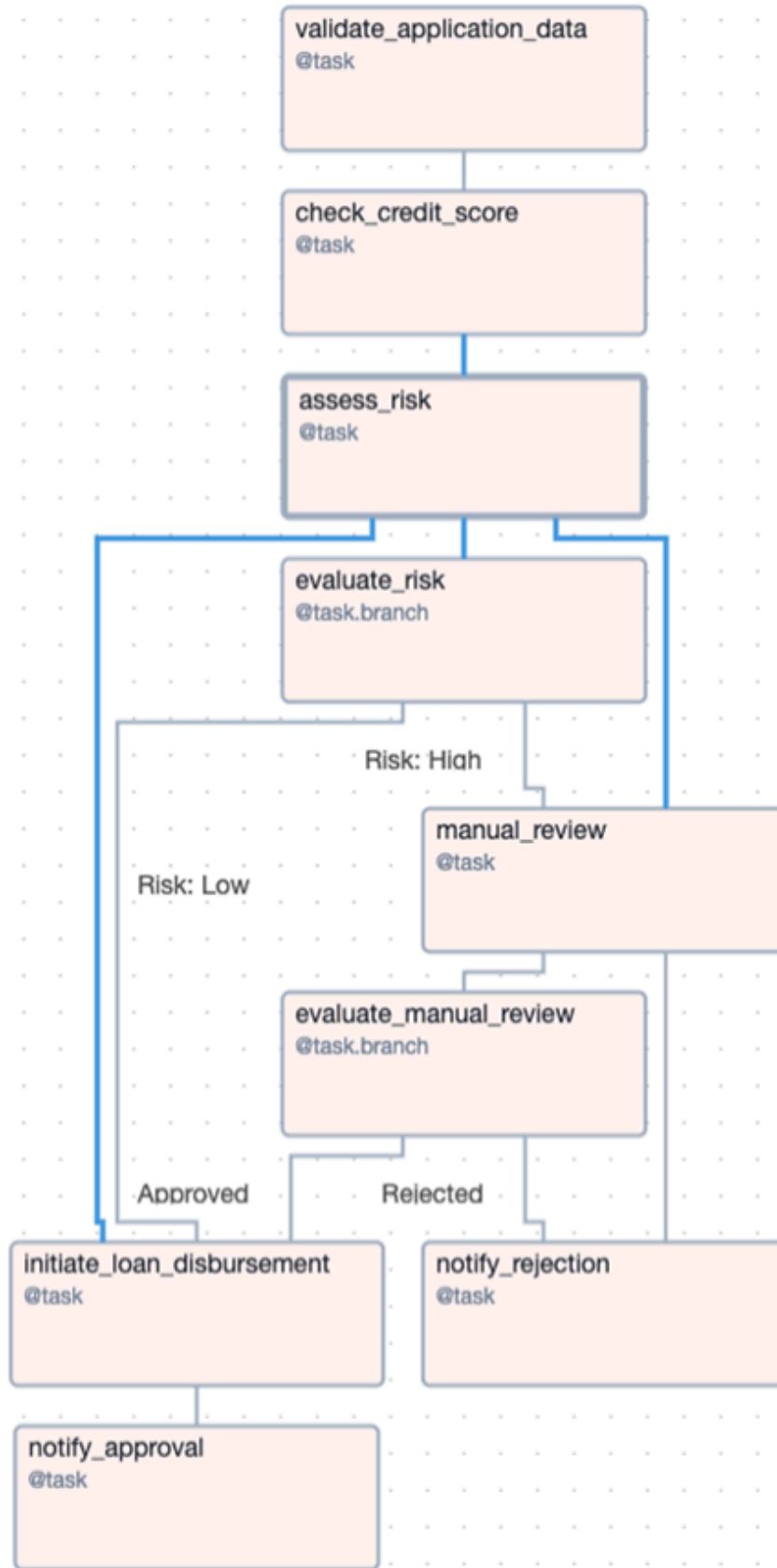
Überlegungen:

- Entwerfen Sie Ihre Zustandsmaschine sorgfältig, um alle möglichen Szenarien zu bewältigen.
- Führen Sie die Schritte nach Möglichkeit parallel durch.
- Verwenden Sie die integrierten Fehlerbehandlungs- und Wiederholungsmechanismen in Step Functions sowohl für permanente als auch für temporäre Fehler.
- Erwägen Sie, je nach Anwendungsfall [Standard- oder Express-Workflows](#) zu verwenden. Express-Workflows sind möglicherweise für Workflows mit kurzer Dauer oder hohem Volumen vorzuziehen.
- [Überwachen Sie die Ausführungskennzahlen](#), um Ihren Workflow zu optimieren.
- Verwenden Sie verschachtelte Workflows, um Funktionen für mehrere Zustandsmaschinen zu kapseln und wiederzuverwenden.
- Für komplexe Workflows sollten Sie die Verwendung von [Amazon Bedrock Agents](#) als Alternative zu Step Functions in Betracht ziehen.

Weitere Informationen finden Sie in der [Dokumentation zu Step Functions](#).

Beispiel: Amazon MWAA

Wenn Ihr Unternehmen Apache Airflow bereits verwendet, ist Amazon MWAA eine natürliche Wahl als Workflow-Orchestrator. In Apache Airflow erstellen Sie Ihre Workflows mithilfe von Python als gerichtete azyklische Graphen (DAGs). Die DAG-Darstellung der Zustandsmaschine, die im Abschnitt Step Functions dargestellt ist, könnte wie folgt aussehen:



Informationen zur Arbeit mit DAGs finden Sie in der [Amazon MWAA-Dokumentation](#).

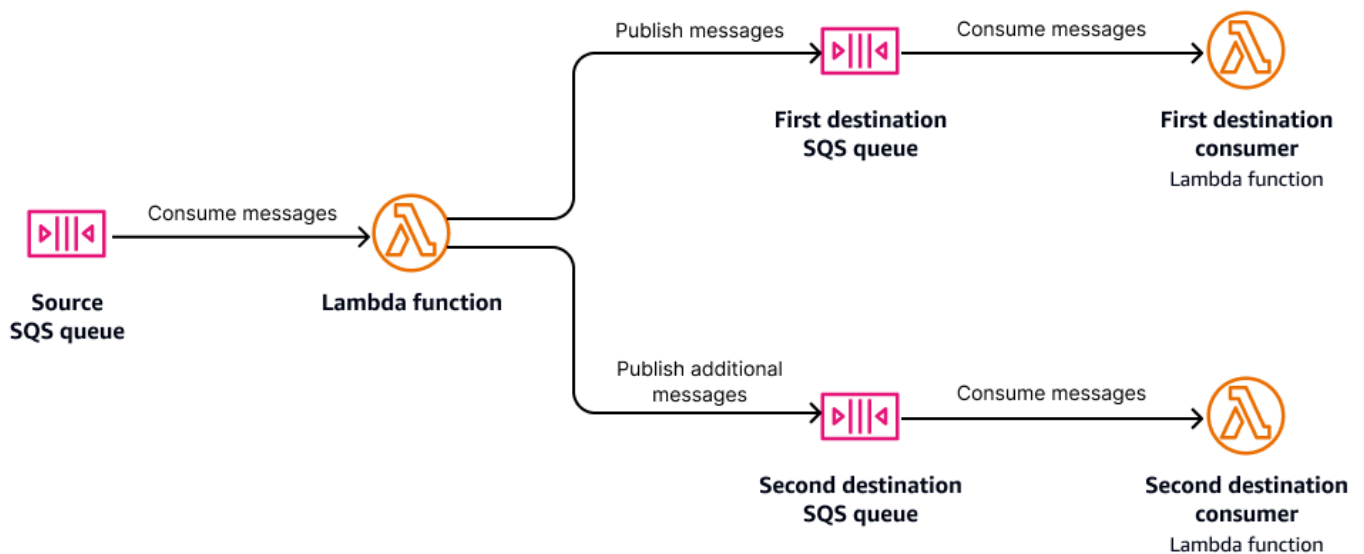
Hauptunterschiede zwischen Step Functions und Amazon MWAA

- Step Functions ist ein vollständig verwalteter serverloser Dienst, sodass keine Infrastruktur vorab bereitgestellt werden muss und kein Wartungsfenster geplant werden muss. Amazon MWAA muss im Voraus bereitgestellt werden, und Sie wählen die Größe und Anzahl der Knoten in Ihrem Cluster.
- In Step Functions können Sie Zustandsmaschinen auf verschiedene Arten erstellen, einschließlich Workflow Studio, direkt als JSON oder mithilfe von AWS Cloud Development Kit (AWS CDK). Apache Airflow DAGs sind in Python geschrieben.
- Mit Step Functions entstehen Ihnen keine Kosten, wenn keine Workflows ausgeführt werden. Mit Amazon MWAA fallen auch dann Kosten an, wenn keine DAGs laufen.

Choreographie

In einem choreografierten System erhalten einzelne Komponenten eine Aufgabe, führen einige Arbeiten aus und geben möglicherweise eine Aufgabe aus, die anschließend ausgeführt werden muss. Es gibt keinen zentralen Orchestrierungsmechanismus. Die Choreographie macht es einfach, Dienste unabhängig voneinander zu skalieren, da jeder Dienst relativ isoliert arbeitet. Er führt Arbeit aus, wenn er Arbeit erhält, und zwar unabhängig davon, zu welchem Durchsatz der Dienst in der Lage ist. Choreographie ist oft ein zentraler Bestandteil einer [ereignisgesteuerten Architektur \(EDA\)](#).

Im folgenden Diagramm gibt es keine Koordination zwischen den Lambda-Funktionen. Jede Funktion verarbeitet nur Nachrichten in der abonnierten Warteschlange. Jede Funktion ist für ihre eigene Fehlerbehandlung verantwortlich und kann die Parallelität steuern, z. B. wenn für eine Downstream-Abhängigkeit ein RPS-Limit (Anfragen pro Sekunde) gilt.



Eine EDA bietet eine Reihe von Vorteilen, wie z. B. eine lose Kopplung von Diensten und Erweiterbarkeit. Eine umfassende Erörterung der EDA-Prinzipien würde den Rahmen dieses Leitfadens sprengen. Weitere Informationen finden Sie unter:

- [AWS Well-Architected Framework — serverlose Anwendungslinse](#)
- [Einführung in die ereignisgesteuerte Architektur \(Serverless Land\)](#)
- [Umstellung auf eine ereignisgesteuerte Architektur \(Serverless Developer Guide\)](#)

Wählen Sie Ihren Koordinationsansatz

Sowohl Choreographie als auch Orchestrierung haben ihren Nutzen bei der Integration von Microservices. Wählen Sie die Choreographie innerhalb der Grenzen eines einzelnen Microservices, bei dem Sie die volle Kontrolle über die Abhängigkeiten haben. Entscheiden Sie sich für die Orchestrierung, wenn Sie über Microservice-Grenzen hinweg arbeiten. Beispielsweise profitieren mehrere Microservices, die an einer verteilten Transaktion beteiligt sind, von der Orchestrierung, um Rollbacks nach Ausfällen zu berücksichtigen. Microservices, die Ereignisse verarbeiten, die für andere Microservices von Interesse sein könnten, werden von der Choreographie und einer ereignisgesteuerten Architektur profitieren.

Ein gängiges Muster für die Implementierung von Rollback, wenn mehrere Systeme an einer einzigen Transaktion beteiligt sind, ist das Saga-Muster.

Verwaltung APIs

Durch ein ordnungsgemäßes API-Management sind Ihre Microservices sowohl für interne als auch für externe Verbraucher zugänglich. AWS bietet eine Vielzahl von Diensten, die Sie zusammen verwenden können, um Ihren APIs Microservice sicher verfügbar zu machen. Diese Dienste ermöglichen es Ihnen, die Sicherheit Ihres Unternehmens durchzusetzen und Überwachung APIs und Beobachtbarkeit von einem zentralen Ort aus zu implementieren. Sie können [Amazon](#) auch verwenden CloudFront, um die Leistung zu verbessern, wenn Ihre Benutzer geografisch weit von dem entfernt APIs sind, in dem AWS-Region die Dienste gehostet werden.

Amazon API Gateway

[Amazon API Gateway](#) ist ein vollständig verwalteter Service, der es Entwicklern ermöglicht, REST in jeder Größenordnung zu erstellen, zu veröffentlichen, zu verwalten, WebSocket APIs zu überwachen und zu sichern. Sie können API Gateway verwenden, um viele der im Abschnitt [Kommunikationsmuster dieses Handbuchs beschriebenen Muster](#) zu implementieren.

Es gibt zwei Haupttypen von REST APIs: REST und HTTP. Beide Typen unterstützen RESTful APIs, bieten aber unterschiedliche Funktionen. Um herauszufinden, was Ihren Anforderungen am besten entspricht, finden [Sie APIs in der API Gateway Gateway-Dokumentation unter Wählen Sie zwischen REST APIs und HTTP](#). Dieser Abschnitt des Handbuchs konzentriert sich auf API Gateway REST APIs.

Die Verwendung von API Gateway als Einstiegspunkt APIs bietet einen zentralen Ort, um allgemeine Anliegen wie Anforderungvalidierung und Sicherheit zu implementieren. API Gateway REST APIs bietet eine [Anforderungvalidierung](#), mit der Sie das Format Ihrer Anfragen mithilfe des [JSON-Schemas](#) definieren können. API Gateway validiert eingehende Anfragen anhand Ihres definierten Schemas und lehnt falsch formatierte Anfragen ab.

Authentifizierung und Autorisierung

API Gateway REST APIs unterstützt die folgenden Authentifizierungs- (AuthN) und Autorisierungsmechanismen (AuthZ):

- IAM — Wenn Sie IAM verwenden, müssen die Anfragen an Ihre API mit [AWS Signature Version 4](#) (Sigv4) signiert werden.
- Amazon Cognito — API Gateway überprüft, ob ein vorgelegtes Inhaber-Token von einem Amazon Cognito Cognito-Benutzerpool ausgestellt wurde. Sie können Ihren Amazon Cognito Cognito-

Benutzerpool auch so konfigurieren, dass er in einen externen Identitätsanbieter (IdP) integriert wird, falls Sie bereits einen verwenden. Sie können auch einen Amazon Cognito Cognito-Benutzerpool für die machine-to-machine (M2M) -Authentifizierung verwenden.

- AWS Lambda authorizer — API Gateway ruft eine Lambda-Funktion auf, die Sie angeben, um alle gewünschten Prüfungen durchzuführen, um festzustellen, ob eine Anfrage autorisiert werden soll.

Weitere Informationen finden Sie unter [Steuern und Verwalten des Zugriffs auf REST APIs](#) in der API Gateway Gateway-Dokumentation.

API-Schlüssel und Ratenbegrenzungen

Mithilfe von API-Schlüsseln APIs und Nutzungsplänen können Sie steuern, wer Sie anrufen darf und zu welchem Tarif. API-Schlüssel sollten nicht zur Authentifizierung verwendet werden, können aber in Verbindung mit den zuvor genannten Schemata verwendet werden. Benutzer müssen nicht immer ihren eigenen API-Schlüssel angeben — Lambda-Autorisierer können beispielsweise einen API-Schlüssel für einen Benutzer zurückgeben. Mit dem Nutzungsplan können Sie den Durchsatz, das Burst-Limit und das monatliche Kontingent angeben. Weitere Informationen finden Sie unter [Nutzungspläne und API-Schlüssel für REST APIs](#) in der API Gateway Gateway-Dokumentation.

Öffentlich und privat APIs

API Gateway REST APIs, auf die über das Internet zugegriffen werden kann, unterstützt zwei Endpunkttypen:

- Edge-optimiert, was bedeutet, dass die Anfragen von Anrufern an einen nahegelegenen CloudFront Point of Presence (POP) weitergeleitet werden. Dies kann zu einer verbesserten Leistung für geografisch verteilte Kunden führen.
- Regional, was bedeutet, dass Anfragen an eine Ressource innerhalb einer bestimmten Ressource weitergeleitet werden. AWS-Region Dies ist eine gute Wahl, wenn sich alle Ihre Kunden in der Nähe der Region befinden, in der Ihre API bereitgestellt wird.

API Gateway REST unterstützt APIs auch private API-Endpunkte, auf die über eine Virtual Private Cloud (VPC) über einen Schnittstellen-VPC-Endpunkt zugegriffen werden kann. Sie können private REST auch sicher teilen, APIs indem Sie VPC-Schnittstellen-Endpunkte in anderen VPCs und sogar anderen erstellen. AWS-Konten Weitere Informationen finden Sie unter [API-Endpunkttypen für REST APIs](#) in der API Gateway Gateway-Dokumentation.

Wann sollte API Gateway verwendet werden

API Gateway ist eine gute Wahl für RESTful Webservices und WebSocket Echtzeitverbindungen. Wenn Sie WebSocket APIs in API Gateway verwenden, können Sie Verhalten für Verbindungs- und Trennungseignisse hinzufügen, z. B. das Speichern von Verbindungen IDs in einem externen Datenspeicher, der mit Clientattributen verknüpft ist. Mithilfe von Nachrichtenattributen können Sie Anfragen auch an benutzerdefinierte Verhaltensweisen weiterleiten.

Sowohl REST als auch WebSocket APIs können direkt integriert werden, AWS-Services ohne dass separate Rechenressourcen wie Lambda-Funktionen erforderlich sind. Dies kann die Leistung verbessern und Kosten senken.

REST APIs unterstützt sowohl pfadbasiertes als auch headerbasiertes Routing, und Sie können sie getrennt oder zusammen verwenden. Ein gängiges Muster besteht darin, eine REST-API als Eingangstür für eine Reihe von Problemen bereitzustellen APIs, gemeinsame Anliegen zu implementieren, sich dann wie ein Reverse-Proxy zu verhalten und autorisierte Anfragen an den richtigen API-Endpunkt weiterzuleiten.

Messaging

Wie im Abschnitt [Kommunikationsmuster](#) beschrieben, können Sie Messaging verwenden, um entweder synchron oder asynchron zwischen Diensten zu kommunizieren. Es stehen viele AWS serverlose Dienste zur Auswahl, und Ihre Auswahl sollte auf Ihren Integrationsanforderungen basieren. Wenn Sie beispielsweise eine bestellte Zustellung von Nachrichten benötigen, sollten Sie einen Service wie Amazon Simple Queue Service (Amazon SQS) oder Amazon Simple Notification Service (Amazon SNS) wählen. Beide Dienste unterstützen die First-in-First-Out-Lieferung (FIFO), im Gegensatz zu Amazon EventBridge, das dies nicht tut.

In den folgenden Abschnitten werden diese Dienste ausführlicher behandelt.

Themen

- [Amazon SQS](#)
- [Amazon SNS](#)
- [Amazon EventBridge](#)
- [AWS AppSync Ereignisse und API Gateway](#)

Amazon SQS

[Amazon SQS](#) unterstützt Standardwarteschlangen, die keine Bestellung garantieren, und FIFO-Warteschlangen, die die Bestellung innerhalb einer bestimmten Nachrichtengruppe garantieren.

Warteschlangen sind eine gängige Methode zur Choreographie von Microservices und bieten dauerhaften Speicherplatz für Nachrichten für bis zu 14 Tage. Warteschlangen werden von Herstellern bevölkert und von Verbrauchern entleert. Wenn Sie sie AWS Lambda als Verbraucher verwenden, können Sie eine SQS-Warteschlange als Ereignisquelle konfigurieren. In diesem Fall fragt das Lambda Service Event Source Mapping (ESM) die Warteschlange für Sie ab und übermittelt Nachrichten an Ihre Lambda-Funktion, sobald sie verfügbar sind. Microservices, die auf anderen Arten von Rechendiensten wie Amazon Elastic Container Service (Amazon ECS) oder Amazon Elastic Compute Cloud (Amazon EC2) ausgeführt werden, müssen ihren eigenen Abfragemechanismus implementieren, um neue Nachrichten aus der Warteschlange abzurufen, sobald sie verfügbar sind.

Lambda ESM für Amazon SQS unterstützt auch Nachrichtenfilterung, sodass Sie nur eine Teilmenge von Nachrichten in einer Warteschlange verarbeiten können, die auf dem Inhalt des Nachrichtentexts basiert.

Abrufen

Amazon SQS unterstützt kurze Abfragen und lange Abfragen von Nachrichten. Short Polling fragt eine Teilmenge von Servern ab, um verfügbare Nachrichten zu finden, und gibt sie sofort zurück. Es werden jedoch möglicherweise nicht alle verfügbaren Nachrichten zurückgegeben. Dies ist nützlich, wenn Ihre Anwendung Nachrichten so schnell wie möglich verarbeiten muss oder längere Wartezeiten nicht tolerieren kann.

Lange Abfragen warten, bis eine konfigurierbare Zeit verstrichen ist oder eine konfigurierbare Anzahl von Nachrichten empfangen wurde, bevor die Nachrichten zurückgegeben werden. Dadurch kann die Anzahl der leeren Umfragen, d. h. die Anzahl der Abfragen, bei denen keine Nachrichten zurückgegeben werden, reduziert werden, insbesondere bei Warteschlangen, die nicht viele Nachrichten empfangen. Durch die Reduzierung der Anzahl leerer Umfragen können Sie Ihre Amazon SQS SQS-Kosten senken, da dieser Service für jede Anfrage Gebühren erhebt und jeder Abfragevorgang eine Anfrage ist.

Empfehlungen

Warteschlangen sind eine gute Wahl, wenn:

- Sie möchten Komponenten entkoppeln und benötigen keine synchrone Kommunikation zwischen ihnen.
- Sie kommunizieren zwischen Komponenten, für die unterschiedliche Verfügbarkeitsvereinbarungen (SLAs) oder Service-Level-Ziele (SLOs) gelten.
- In der Regel haben Sie einen einzigen Benutzer für eine Reihe von Nachrichten.

Erwägen Sie eine alternative Option, wenn:

- Sie benötigen synchrone Kommunikation.
- Sie benötigen eine komplizierte Routing-Logik, um Nachrichten an den richtigen Verbraucher zu senden.

Amazon SNS

Mit [Amazon SNS](#) können Sie sowohl Standard- als auch FIFO-Themen erstellen. Themen werden verwendet, um eine publish/subscribe (Pub/Sub-) Architektur zu implementieren. Amazon SNS unterstützt eine Vielzahl von Abonnementtypen, darunter E-Mail, SMS (vorausgesetzt, Sie haben eine Ausgangsidentität konfiguriert, z. B. eine gebührenfreie Nummer oder einen 10-stelligen Langcode), HTTP (S) -Endpunkte und SQS-Warteschlangen. Endbenutzer-Abonnements, wie E-Mail-Nachrichten und SMS-Nachrichten, für ein SNS-Thema müssen vom Abonnenten bestätigt werden. Amazon SNS ermöglicht es Diensten, sich weit auszubreiten, was bedeutet, dass eine einzelne Nachricht an eine potenziell große Anzahl von Abonnenten zugestellt werden kann. Ein SNS-Standardthema hat ein Standardlimit von 12,5 Millionen Abonnements.

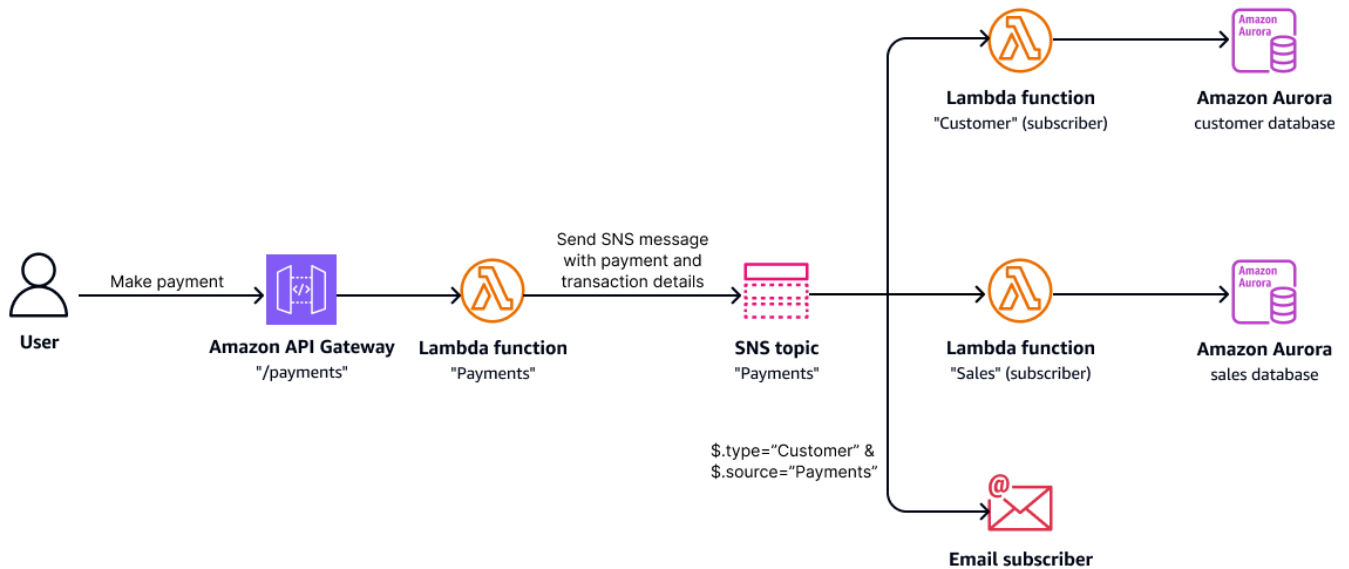
In einer Microservices-Umgebung ist ein SNS-Thema nützlich, um die Routing- und Übermittlungslogik von der Logik eines Herausgebers zu entkoppeln. Dies kann mithilfe von Themenfiltern implementiert werden. Konzeptionell sind Themenfilter den EventBridge Amazon-Regeln etwas ähnlich, sie werden jedoch für jeden Abonnenten konfiguriert, anstatt von einem zentralen Ort aus verfügbar zu sein. Nehmen wir zum Beispiel an, Sie haben:

- Ein Bestellservice, der Bestellungen bearbeitet.
- Ein Fulfillment-Service, der die Auftragserfüllung abwickelt.
- Ein Treuedienst, der Mitgliedern Treuepunkte für Bestellungen vergibt.

Wenn eine Bestellung zur Ausführung bereit ist, veröffentlicht sie eine Nachricht zu einem Thema. Der Fulfillment-Service abonniert das Thema, wendet aber keinen Filter an, da er über alle Bestellungen informiert werden möchte. Stellen Sie sich vor, Sie haben einen Treueservice, der dafür verantwortlich ist, Mitgliedern Punkte zu gewähren, wenn sie eine Bestellung aufgeben. Es werden jedoch nicht alle Bestellungen von Mitgliedern aufgegeben. Der Treueservice würde das Thema abonnieren, würde aber einen Abonnementfilter implementieren, um ein Attribut zu überprüfen, das angibt, ob die Bestellung für ein Mitglied oder einen Gast war.

Stellen Sie sich den Fall vor, dass ein System eine Zahlungsaufforderung von einem Endbenutzer erhält, wie in der folgenden Abbildung dargestellt. In diesem Fall müssen mehrere nachgelagerte Systeme wissen, dass die Anfrage gestellt wurde, damit verschiedene Maßnahmen ergriffen werden können. Wenn Sie Amazon SNS verwenden, werden Zahlungen unter einem SNS-Thema veröffentlicht, und Lambda-Funktionen abonnieren das Thema, um die Kunden- und Vertriebsdatenbanken zu aktualisieren. Darüber hinaus sendet ein E-Mail-Abonnement (das vom

Kunden bestätigt werden muss) mithilfe eines Abonnementfilters eine E-Mail-Bestätigung an den Kunden.



Empfehlungen

Einige der in diesem Abschnitt für Amazon SNS beschriebenen Funktionen überschneiden sich mit den Funktionen, die ein Event-Bus bietet, wie z. B. EventBridge. Erwägen Sie die Verwendung von Amazon SNS in folgenden Fällen:

- Sie werden eine große Anzahl von Abonnenten für ein Thema haben.
- Sie möchten Abonnementtypen (wie E-Mail oder SMS) verwenden, die nicht nativ von EventBridge unterstützt werden.
- Abonnenten sollten in der Lage sein, ihren Abonnementfilter zu bestimmen.
- Sie benötigen eine geordnete Zustellung an Abonnenten (pro Nachrichtengruppe).

Wenn Sie viele Themen haben und Abonnements und Filter verwendet werden, um Nachrichten zwischen Microservices weiterzuleiten, ist EventBridge wahrscheinlich die bessere Wahl.

Amazon EventBridge

[Amazon EventBridge](#) ist ein serverloser Event-Bus-Service und dient häufig als Grundlage für eine ereignisgesteuerte Architektur (EDA). Sie können ihn auch verwenden, um Nachrichten asynchron

zwischen Microservices weiterzuleiten und zu übermitteln. Mithilfe dieser EventBridge Methode veröffentlichen Produzenten Ereignisse auf einem Bus. Sie konfigurieren Regeln, die Ereignisse anhand ihres Inhalts zuordnen, und wählen ein oder mehrere Ziele aus, an die Ereignisse, die dieser Regel entsprechen, übermittelt werden sollen. EventBridge unterstützt eine [Vielzahl von Regelzielen](#). Mithilfe eines Event-Busses können Sie Produzenten von Verbrauchern trennen und Ihre Routing- und Lieferlogik konsolidieren.

EventBridgeIn können Sie auch geplante Regeln erstellen, sodass Sie zu bestimmten Zeiten Aktionen ergreifen können. Sie können Ereignisse mithilfe von cron-basierten und ratenbasierten Ausdrücken definieren.

[EventBridgePipes](#) bietet Ihnen die Möglichkeit, Nachrichten von einer [Quelle](#) an ein [Ziel](#) weiterzuleiten, ohne einen Rechendienst wie z. AWS Lambda Nehmen wir zum Beispiel an, Sie haben eine SQS-Warteschlange, die Nachrichten empfängt, was eine AWS Step Functions Zustandsmaschine auslösen sollte. Anstatt eine Lambda-Funktion zu erstellen, die über eine Ereignisquellenzuordnung verfügt, um Nachrichten aus der Warteschlange zu verarbeiten, und Code zu schreiben, indem Sie ein AWS SDK verwenden, um die Zustandsmaschine aufzurufen, können Sie EventBridge Pipes verwenden, um dies für Sie zu tun, ohne benutzerdefinierten Code zu schreiben.

EventBridge wird häufig mit anderen Messaging-Diensten wie Amazon SQS und Amazon SNS verwendet. Durch die Übermittlung von Ereignissen an eine SQS-Warteschlange erhält ein Empfangsdienst beispielsweise die Flexibilität, Nachrichten zu verarbeiten, wenn er dazu in der Lage ist, und zwar mit einer Geschwindigkeit, die unabhängig von der Geschwindigkeit ist, mit der Ereignisse erzeugt werden. Ebenso können Sie Ereignisse übermitteln, die einer großen Anzahl von Abonnenten eines SNS-Themas zugänglich gemacht werden sollen.

Empfehlungen

Verwenden Sie, wenn: EventBridge

- Sie benötigen keine synchrone Kommunikation zwischen Diensten.
- Sie möchten die Nachrichtenrouting-Logik von Ihren Microservices entkoppeln. Die Microservices erzeugen nur Ereignisse und veröffentlichen sie im Event-Bus. Interessierte Dienste erstellen Regeln, um diese Ereignisse abzugleichen und bereitzustellen.
- Sie müssen Nachrichten von einem unterstützten Dienst an einen anderen übermitteln.

Ziehen Sie andere Dienste in Situationen in Betracht, in denen:

- Sie benötigen eine strikte Reihenfolge der Ereignisse. In diesen Fällen sollten Sie Amazon SQS FIFO-Warteschlangen oder Amazon SNS FIFO-Themen in Betracht ziehen. Ziehen Sie alternativ Event-Stream-Services wie Amazon Kinesis Data Streams oder Amazon Managed Streaming for Apache Kafka (Amazon MSK) in Betracht.

AWS AppSync Ereignisse und API Gateway

AWS AppSync Events und Amazon API Gateway bieten beide ein verwaltetes WebSocket Erlebnis für Ihre Microservices.

[AWS AppSync Events](#) bietet eine vereinfachte Erfahrung für Echtzeit-Messaging mithilfe von. WebSocket AWS AppSync Events unterstützt Unicast- und Multicast-Messaging sowie eine flexible Gruppierung von Kanälen in Namespaces mit Unterstützung für Platzhalter. Microservices können mithilfe von Events auf unterschiedliche Weise miteinander kommunizieren. AWS AppSync Beispielsweise kann ein Dienst, der Echtzeitdaten empfängt, die Daten transformieren und auf dem entsprechenden Kanal veröffentlichen, wo Abonnenten sie in Echtzeit erhalten.

[API Gateway](#) unterstützt auch WebSocket APIs. Sie können Integrationen mit AWS-Services, wie Amazon DynamoDB, definieren AWS Lambda und Routenauswahlausdrücke konfigurieren, die diesen Integrationen zugeordnet sind. API Gateway verfügt über spezielle Routen, mit denen Sie Ihre WebSocket Verbindungen autorisieren und verwalten können. Je nach Ihren Anforderungen können Sie WebSocket Verbindungsinformationen in einem Datenspeicher wie DynamoDB speichern. Mithilfe dieser Informationen können Nachrichten mit einer bestimmten WebSocket Verbindungs-ID über eine REST-API für bestimmte Verbindungen veröffentlicht werden.

Empfehlungen

Verwenden Sie AWS AppSync Ereignisse, wenn:

- Sie haben mehrere Messaging-Kanäle, die in Namespaces gruppiert sind, und möchten mithilfe von Platzhaltern Kanalgruppen veröffentlichen und abonnieren.
- Ihre Kommunikation findet hauptsächlich zwischen verschiedenen Systemen statt. AWS-Services

Verwenden Sie API Gateway WebSocket APIs , wenn:

- Sie möchten es Clients ermöglichen, in Echtzeit persistente Verbindungen zu AWS-Service Integrationen herzustellen.

- Sie möchten die WebSocket Verbindungen selbst verwalten. Möglicherweise möchten Sie anderen Systemen erlauben, Nachrichten an einen bestimmten Client zu senden, nachdem sie deren Verbindungs-ID nachgeschlagen haben.
- Sie möchten API Gateway Gateway-Funktionen wie Stage-Deployments oder Proxy-Integrationen verwenden, oder Sie möchten Ihre eigenen Subprotokolle konfigurieren.

Häufig gestellte Fragen

Wie kann ich verschiedene Integrationsmuster kombinieren?

In den meisten Situationen werden Sie Integrationsmuster kombinieren wollen. Sie können dies beispielsweise verwenden, um einen Prozess AWS Step Functions zu orchestrieren, der einen Remotedienst aufruft, indem Sie das Claim Check-Muster verwenden. Oder Sie haben vielleicht einen orchestrierten Prozess, der Nachrichten in Warteschlangen platziert, was wiederum choreografierte Dienste auslöst.

Was ist der Hauptvorteil der Verwendung einer Microservices-Architektur?

Zu den Hauptvorteilen gehören die unabhängige Skalierung von Diensten, eine verbesserte Fehlerisolierung, eine höhere Entwicklungsgeschwindigkeit durch parallel Teamarbeit und die Fähigkeit zur kontinuierlichen Bereitstellung und Bereitstellung (CI/CD).

Wie kann ich die Fehlerbehandlung in diesen Mustern implementieren?

Sie können die Fehlerbehandlung mithilfe der integrierten Mechanismen in implementieren AWS-Services. AWS Lambda Funktionen können beispielsweise mit Wiederholungslogik konfiguriert werden, und Amazon SQS unterstützt Warteschlangen mit unerlaubten Buchstaben für die Behandlung anhaltender Fehler. Darüber hinaus bietet Step Functions Fehlerbehandlungs- und Wiederholungsmechanismen auf Workflow-Ebene.

Was sind die Vorteile der Verwendung des Anspruchsprüfmusters bei asynchroner Kommunikation?

Das Muster zur Prüfung von Ansprüchen ermöglicht es Kunden, bei Einreichung der Anfrage eine Kennung zu erhalten. Diese Kennung kann später verwendet werden, um den Status zu überprüfen und das Ergebnis abzurufen. Dieses Muster kommt Kunden zugute, da es einen Mechanismus bietet, mit dem Ergebnisse abgefragt werden können, ohne synchron warten zu müssen. Weitere Informationen finden Sie im Abschnitt [Prüfung von Ansprüchen](#) weiter oben in diesem Handbuch.

Wie verbessert das Callback-Muster die asynchrone Kommunikation in Microservices?

Das Rückrufmuster verbessert die asynchrone Kommunikation, indem es dem Client ermöglicht, einen Standort anzugeben, an den der Dienst nach Abschluss der Verarbeitung Kontakt aufnehmen kann. Dies entkoppelt den Client vom Warten auf eine Antwort und ermöglicht es ihm, mit anderen Aufgaben fortzufahren. Weitere Informationen finden Sie im Abschnitt [Rückruf](#) weiter oben in diesem Handbuch.

Kann ich mithilfe der beschriebenen Muster bidirektionale Kommunikation in Microservices implementieren?

Sie können bidirektionale Kommunikation implementieren, indem Sie eine statusbehaftete Verbindung zwischen einem Client und einem Dienst herstellen, sodass beide Nachrichten asynchron senden und verarbeiten können. Dazu muss der Dienst eine offene Verbindung für jeden Client unterstützen. Weitere Informationen finden Sie im Abschnitt [Bidirektionale Kommunikation](#) weiter oben in diesem Handbuch.

Wie kann ich die Verwendung von Lambda-Funktionen in asynchronen Kommunikationsmustern optimieren?

Sie können Lambda-Funktionen optimieren, indem Sie sicherstellen, dass sie idempotent sind, um potenzielle Nachrichtenduplikationen zu bewältigen, indem Sie Amazon SQS SQS-Funktionen wie Nachrichtengruppen für die Bestellung verwenden und lange Abfragen implementieren, um die Kosten zu senken. Darüber hinaus können Sie Ausführungskennzahlen überwachen, um Optimierungsmöglichkeiten zu identifizieren.

Was sind die Hauptunterschiede zwischen der Verwendung von Amazon SNS und EventBridge für das pub/sub Muster?

Amazon SNS sendet eine einzige Nachricht an alle Abonnenten, die für einige Abonnenten möglicherweise unnötige Daten enthält. Amazon EventBridge ermöglicht eine detailliertere Steuerung, da Sie mehrere Regeln verwenden können, die einem einzelnen Ereignis entsprechen, wobei jede Regel einen anderen nachgelagerten Service oder eine andere nachgelagerte Aktion

auslöst. Weitere Informationen finden Sie in [Amazon SNS](#) und in den [EventBridge](#) Abschnitten weiter oben in diesem Handbuch.

Ressourcen

AWS-Service Dokumentation

- [Amazon API Gateway](#)
- [AWS AppSync -Ereignisse](#)
- [Amazon EventBridge](#)
- [Amazon MWA](#)
- [Amazon SNS](#)
- [Amazon SQS](#)
- [AWS Step Functions](#)

Zusätzliche Lektüre

- [Strategie zur Modernisierung von Anwendungen in der AWS Cloud](#)
- [Stufenweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud](#)
- [Bewertung der Modernisierungsbereitschaft von Anwendungen in der AWS Cloud](#)
- [Zerlegung von Monolithen in Microservices](#)
- [Implementierung von Integrationsmustern für Unternehmen mit AWS Messaging-Diensten: Kanäle point-to-point](#)
- [Pub/Sub-Messaging: Asynchrone Ereignisbenachrichtigungen](#)

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
Wichtige Aktualisierungen	Erweiterter, neu organisierter und aktualisierter Leitfaden , um AWS-Service Aktualisierungen widerzuspiegeln.	10. September 2025
Erste Veröffentlichung	—	11. Januar 2021

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:

- **Refactor/re-architect** — Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile der Cloud-nativen Funktionen 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-Compatible 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 Kundenbeziehungsmanagement (CRM) -System 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

A2A () Agent-to-Agent

Ein Stateful-Protokoll für die Zusammenarbeit zwischen Agenten, das die Delegation von Aufgaben und die Zustandsübertragung unterstützt.

ABAC

Siehe [attributbasierte Zugriffskontrolle](#).

abstrahierte Dienste

Siehe [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 verbindenden Anwendungen, während Daten in die Zieldatenbank repliziert werden. Die Zieldatenbank akzeptiert während der Migration keine Transaktionen.

Agent

Ein KI-System, das mithilfe von Tools selbständig Überlegungen anstellen, planen und Maßnahmen ergreifen kann, um Ziele zu erreichen.

Agent Ops

Operative Verfahren zum Erstellen, Testen, Bereitstellen und Ausführen von KI-Agenten in der Produktion im großen Maßstab.

Aggregatfunktion

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

AI

Siehe [künstliche Intelligenz](#).

AIOps

Siehe [Operationen mit künstlicher Intelligenz](#).

Anonymisierung

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

Anti-Muster

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

Anwendungssteuerung

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

Anwendungsportfolio

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

künstliche Intelligenz (KI)

Das Gebiet der Datenverarbeitungswissenschaft, das sich der Nutzung von Computertechnologien zur Ausführung kognitiver Funktionen widmet, die typischerweise mit

Menschen in Verbindung gebracht werden, wie Lernen, Problemlösen und Erkennen von Mustern. Weitere Informationen finden Sie unter [Was ist künstliche Intelligenz?](#)

Operationen mit künstlicher Intelligenz (AIOps)

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

Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

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

Attributbasierte Zugriffskontrolle (ABAC)

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

autoritative Datenquelle

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

Availability Zone

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

AWS Framework für die 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 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.

blue/green Einsatz

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, sogenannte bösartige Bots, 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 in den Leitlinien unter dem Indikator „[Glasbruchverfahren implementieren](#)“. AWS Well-Architected

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 [AWS Framework 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 Störungsereignissen, um die Widerstandsfähigkeit eines Systems zu testen. Sie können [AWS Fault Injection Service \(AWS FIS\)](#) verwenden, um Experimente durchzuführen, die Ihre AWS Workloads stress, und deren Reaktion zu bewerten.

CI/CD

Siehe [Continuous Integration und Continuous Delivery](#).

Klassifizierung

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

Citizen Developer

Ein Geschäftsanwender, der KI-Anwendungen mithilfe von Plattformen ohne Programmierkenntnisse erstellt. code/low

clientseitige Verschlüsselung

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

Cloud-Kompetenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte

Transformationen. Weitere Informationen finden Sie in den [CCoE-Beiträgen](#) im AWS Cloud Enterprise Strategy Blog.

Cloud Computing

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

Cloud-Betriebsmodell

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

Phasen der Einführung der Cloud

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

- Projekt – Durchführung einiger Cloud-bezogener Projekte zu Machbarkeitsnachweisen und zu Lernzwecken
- Fundament – Grundlegende Investitionen tätigen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer Landing Zone, Definition eines CCoE, Einrichtung eines Betriebsmodells)
- Migration – Migrieren einzelner Anwendungen
- Re-invention — Optimierung von Produkten und Dienstleistungen sowie 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 Repositorys 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 mehr richtlinienkonform wird, und zwar in der Regel schrittweise und unbeabsichtigt.

Verwaltung der Datenbankkonfiguration (CMDB)

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

Konformitätspaket

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

kontinuierliche Integration und kontinuierliche Bereitstellung () 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 des AWS Well-Architected Frameworks. 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.

betreffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

Ein Datenverwaltungssystem, das Business Intelligence wie Analysen unterstützt. Data Warehouses enthalten in der Regel große Mengen historischer 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.

Tiefgreifende Verteidigung

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 umfassender Verteidigungsansatz könnte beispielsweise Multi-Faktor-Authentifizierung, Netzwerksegmentierung und Verschlüsselung kombinieren.

delegierter Administrator

Ein kompatibler Dienst ein AWS Mitgliedskonto registrieren AWS Organizations, um die Konten der Organisation zu verwalten 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 Sicherheitsereignissen, bei denen die vorhandenen

präventiven Kontrollen umgangen wurden. Weitere Informationen finden Sie unter [Detektivische Kontrolle](#) in Implementierung von Sicherheitskontrollen in AWS.

Abbildung des Wertstroms in der Entwicklung (DVSM)

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

digitaler Zwilling

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

Maßtabelle

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

Katastrophe

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

Disaster Recovery (DR)

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

DML

Siehe [Sprache zur Datenbankmanipulation](#).

Domainorientiertes Design

Ein Ansatz zur Entwicklung eines komplexen Softwaresystems, bei dem seine Komponenten mit sich entwickelnden Domains oder Kerngeschäftsziele verknüpft werden, denen jede

Komponente dient. Dieses Konzept wurde von Eric Evans in seinem Buch Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003) vorgestellt. Informationen darüber, wie Sie domänengesteuertes Design mit dem Strangler-Fig-Muster verwenden können, finden Sie unter Schrittweise [Modernisierung 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 der Wertströme 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 niedrigstwertige Byte zuerst.

Endpunkt

Siehe [Service-Endpunkt](#).

Endpunkt-Services

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

Unternehmensressourcenplanung (ERP)

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

Envelope-Verschlüsselung

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

Umgebung

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

- Entwicklungsumgebung – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere

Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.

- Niedrigere Umgebungen – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.
- Produktionsumgebung – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

Epics

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

ERP

Siehe [Enterprise Resource Planning](#).

Explorative Datenanalyse (EDA)

Der Prozess der Analyse eines Datensatzes, um seine Hauptmerkmale zu verstehen. Sie sammeln oder aggregieren Daten und führen dann erste Untersuchungen durch, um Muster zu finden, Anomalien zu erkennen und Annahmen zu überprüfen. EDA wird durchgeführt, indem zusammenfassende Statistiken berechnet und Datenvisualisierungen erstellt werden.

F

Faktentabelle

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

schnell scheitern

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

Grenze zur Fehlerisolierung

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

Feature-Zweig

Siehe [Zweig](#).

Features

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

Bedeutung der Feature

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

Featuretransformation

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

Eingabeaufforderung mit wenigen Klicks

Bereitstellung einer kleinen Anzahl von Beispielen, die die Aufgabe und das gewünschte Ergebnis veranschaulichen, bevor das [LLM](#) 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. Few-shot Eingabeaufforderungen können bei Aufgaben, die spezifische Formatierungs-, Argumentations- oder Fachkenntnisse erfordern, effektiv sein. Siehe auch [Zero-Shot-Eingabeaufforderung](#).

FGAC

Siehe [detaillierte 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](#).

FM-Gateway

Ein zentraler Vermittler, der den Zugriff auf Basismodelle kontrolliert und normalisiert. Wird auch als LLM-Gateway bezeichnet.

G

Generative KI

Eine Untergruppe von [KI-Modellen](#), die mit großen Datenmengen trainiert wurden und mithilfe 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 dabei hilft, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Organisationseinheiten (OUs) zu regeln. Präventiver Integritätsschutz setzt Richtlinien durch, um die Einhaltung von Standards zu gewährleisten. Sie werden mithilfe von Service-Kontrollrichtlinien und IAM-Berechtigungsgrößen 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.

Leitplanken (KI)

Sicherheitsmechanismen, die Eingaben und Ausgaben von [Agenten](#) filtern, validieren und einschränken, um ein verantwortungsbewusstes und sicheres Verhalten der KI zu gewährleisten.

H

HEKTAR

Siehe [Hochverfügbarkeit](#).

Heterogene Datenbankmigration

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

hohe Verfügbarkeit (HA)

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

historische Modernisierung

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

Holdout-Daten

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 Modelleleistung zu bewerten, indem Sie die Modellvorhersagen mit den Holdout-Daten vergleichen.

Der Mensch im Kreis (HiTL)

Ein Workflow-Muster, bei dem die Ausführung von [Agenten an kritischen](#) Entscheidungspunkten unterbrochen wird, um von einem Mitarbeiter geprüft und genehmigt zu werden.

Homogene Datenbankmigration

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

heiße Daten

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

Hotfix

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

Hypercare-Phase

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

I

IaC

Sehen Sie sich [Infrastruktur als Code](#) an.

Identitätsbasierte Richtlinie

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

Leerlaufanwendung

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

IloT

Siehe [Industrielles Internet der Dinge](#).

unveränderliche Infrastruktur

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

Eingehende (ingress) VPC

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

Inkrementelle Migration

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

Industrie 4.0

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

Infrastruktur

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

Infrastructure as Code (IaC)

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

Industrielles Internet der Dinge (IIoT)

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

Inspektions-VPC

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

Internet of Things (IoT)

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

Interpretierbarkeit

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

T service management (ITSM, IT-Service management)

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

Siehe [IT-Informationsbibliothek](#).

ITSM

Siehe [IT-Servicemanagement](#).

L

Labelbasierte Zugangskontrolle (LBAC)

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

Landing Zone

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

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.

LBAC

Siehe [Labelbasierte 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 Verfolgung, Überwachung, Dokumentation und Steuerung von Produktionsprozessen, bei denen Rohstoffe in der Fertigung zu fertigen Produkten umgewandelt werden.

MAP

Siehe [Migration Acceleration Program](#).

MCP

Siehe [Model Context Protocol](#).

Model Context Protocol (MCP)

[Ein zustandsloses Protokoll für die Kommunikation zwischen Agenten und Tool.](#)

MCP-Server

Ein Dienst, der ein oder mehrere [Tools](#) über das [Model Context](#) Protocol verfügbar macht.

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 [Mechanismen](#) im AWS Well-Architected Framework erstellen.

Mitgliedskonto

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

MES

Siehe [Manufacturing Execution System](#).

Message Queuing-Telemetrietransport (MQTT)

[Ein leichtes, auf dem publish/subscribeMuster basierendes M2M-Kommunikationsprotokoll \(Machine-to-Machine\) für IoT-Geräte mit beschränkten Ressourcen.](#)

Microservice

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

Microservices-Architekturen

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

Cross-functional Teams, die die Migration von Workloads durch automatisierte, agile Ansätze optimieren. Zu den Teams von Migration Factory gehören in der Regel Betriebsanalysten 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. Um die Konsistenz, Zuverlässigkeit und Vorhersagbarkeit zu verbessern, empfiehlt das AWS Well-Architected Framework die Verwendung einer [unveränderlichen Infrastruktur](#) als bewährte Methode.

O

OAC

Siehe [Origin Access Control](#).

EICHE

Siehe [Zugriffsidentität von Origin](#).

COM

Siehe [organisatorisches Change-Management](#).

Offline-Migration

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

OI

Siehe [Betriebsintegration](#).

OLA

Siehe Vereinbarung auf [operativer Ebene](#).

Online-Migration

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

OPC-UA

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

Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

Ein Machine-to-Machine-Kommunikationsprotokoll (M2M) 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 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.

Ursprungszugriffsidentität (OAI)

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

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 -Referenzarchitektur für die Sicherheit](#) empfiehlt, Ihr Netzwerkkonto mit eingehenden und ausgehenden VPCs und Inspektions-VPCs einzurichten, um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet zu schützen.

P

Berechtigungsgrenze

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

persönlich identifizierbare Informationen (PII)

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

Personenbezogene Daten

Siehe [persönlich identifizierbare Informationen](#).

Playbook

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

PLC

Siehe [programmierbare Logiksteuerung](#).

PLM

Siehe [Produktlebenszyklusmanagement](#).

policy

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

Polyglotte Beharrlichkeit

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

Portfoliobewertung

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

predicate

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

Prädikat Pushdown

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

Präventive Kontrolle

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

Prinzipal

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

Datenschutz 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 ihre Subdomains innerhalb einer oder mehrerer VPCs reagieren soll. Weitere Informationen finden Sie unter [Arbeiten mit privat gehosteten Zonen](#) in der Route-53-Dokumentation.

proaktive Steuerung

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

Produktlebenszyklusmanagement (PLM)

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

Produktionsumgebung

Siehe [Umgebung](#).

Speicherprogrammierbare Steuerung (SPS)

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

schnelle Verkettung

Verwenden Sie die 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 \(RACI\)](#).

RAG

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.

Refaktorisierung

Siehe [7 Rs](#).

Region

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

Regression

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

rehosten

Siehe [7 Rs.](#)

Veröffentlichung

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

umziehen

Siehe [7 Rs.](#)

neue Plattform

Siehe [7 Rs.](#)

Rückkauf

Siehe [7 Rs.](#)

Ausfallsicherheit

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

Ressourcenbasierte Richtlinie

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

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

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

Reaktive Kontrolle

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

Beibehaltung

Siehe [7 Rs](#).

zurückziehen

Siehe [7 Rs](#).

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 Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: [präventiv](#), [detektiv](#), [reaktionsschnell](#) und [proaktiv](#).

Härtung der Sicherheit

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

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

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

Automatisierung von Sicherheitsreaktionen

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

Serverseitige Verschlüsselung

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

Service-Kontrollrichtlinie (SCP)

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

Service-Endpunkt

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

Service Level Agreement (SLA)

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

Service-Level-Indikator (SLI)

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

Service-Level-Ziel (SLO)

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

Modell der geteilten Verantwortung

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

Schatten-KI

Nicht autorisierte [KI-Anwendungen](#), die außerhalb der kontrollierten Kanäle innerhalb eines Unternehmens erstellt oder verwendet wurden.

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 [Schrittweise Modernisierung 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

Key-value Paare, die als Metadaten für die Organisation Ihrer AWS Ressourcen dienen. 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.

tool

Eine Funktion oder API, die ein [Agent](#) aufrufen kann, um Operationen in externen Systemen auszuführen.

Transit-Gateway

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

Stammbasierter Workflow

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

Vertrauenswürdiger Zugriff

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

Optimieren

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

Zwei-Pizzen-Team

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

U

Unsicherheit

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

undifferenzierte Aufgaben

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

höhere Umgebungen

Siehe [Umgebung](#).

V

Vacuuming

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

Versionskontrolle

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

VPC-Peering

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

Schwachstelle

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

W

Warmer Cache

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

warme Daten

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

Fensterfunktion

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

Workload

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

Workstream

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

WURM

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

Eingabeaufforderung ohne Vorwarnung

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.