



Umgestaltung Ihrer Mainframe-Anwendungen mithilfe einer gemeinsamen IBM Db2 for z/OS-Datenbank für eine schrittweise Migration

AWS Präskriptive Leitlinien



AWS Präskriptive Leitlinien: Umgestaltung Ihrer Mainframe- Anwendungen mithilfe einer gemeinsamen IBM Db2 for z/OS-Datenbank für eine schrittweise Migration

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

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

Table of Contents

Einführung	1
Geschäftsergebnisse	2
AWS Mainframe Modernization	4
Plattformwechsel	4
Automatisiertes Refactoring	5
Vorteile von Replatforming	5
Der Transformationsprozess	7
Planung	8
Entdeckung von Anwendungen	8
Abhängigkeiten von Daten	8
Kapazitäts-Benchmark	9
Wellenplanung	12
Erstellung	12
Konsistenz der Anwendungen	13
Architektur	14
In Ausführung	16
Zweiphasiges Commit (2 PC)	17
Laufzeit-Infrastruktur	18
Testen	19
Quellumgebung	19
Zielumgebung	20
Analyse	20
Testen Sie Ihre Anwendung in AWS Mainframe Modernization	21
Cutover	23
Bereitstellung	23
Gehen Sie live	24
Rollback	24
Schließen Sie ab	24
Architektur	24
Bewährte Methoden	26
Netzwerklatenz	26
Sicherheit	27
Anwendungs-Governance	27
Elastizität	28

Nächste Schritte	29
Ressourcen	31
AWS Dokumentation	31
Referenzen von Rocket Software	31
IBM-Referenzen	31
Tools	31
AWS Präskriptive Leitlinien und Leitfäden	31
Dokumentverlauf	32
Glossar	33
#	33
A	34
B	37
C	39
D	43
E	47
F	49
G	51
H	52
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

Umgestaltung Ihrer Mainframe-Anwendungen mithilfe einer gemeinsamen IBM Db2-Datenbank für die schrittweise Migration z/OS

Luis Gustavo Dantas und André Botura, Amazon Web Services (AWS)

Mai 2025 ([Geschichte des Dokuments](#))

In der sich ständig verändernden Landschaft der Unternehmenstechnologie ist die Modernisierung von Mainframes zu einer wichtigen Anforderung für Unternehmen geworden, die wettbewerbsfähig und agil bleiben müssen. Bei dieser Transformation geht es nicht nur darum, alte Systeme durch neue zu ersetzen, sondern es ist eine strategische Weiterentwicklung, die die Lücke zwischen den robusten, zuverlässigen Grundlagen der Vergangenheit und den dynamischen, innovativen Möglichkeiten der future schließt.

Der Mainframe, einst der unangefochtene Marktführer im Bereich Enterprise Computing, befindet sich jetzt an einem Wendepunkt. Seine unvergleichliche Rechenleistung und seine Sicherheitsfunktionen haben ihn jahrzehntelang relevant gemacht, aber die heutigen Unternehmen benötigen Systeme, die sich nahtlos in Cloud-Dienste integrieren lassen, mobile Anwendungen unterstützen und das Potenzial künstlicher Intelligenz und Big-Data-Analysen nutzen können.

Modernisierung erfordert nicht immer eine vollständige Migration weg von Mainframes. Einige Unternehmen entscheiden sich für hybride Ansätze, die die Stärken von Mainframe- und Cloud-Umgebungen nutzen. Diese Strategie ermöglicht es ihnen, wichtige Legacy-Anwendungen beizubehalten und gleichzeitig schrittweise auf modernere Plattformen umzusteigen. Dieser technologische Wandel beinhaltet mehr als nur Systemaktualisierungen. Er erfordert eine Veränderung der Unternehmenskultur und der Fähigkeiten. Im Zuge der Modernisierung investieren Unternehmen sowohl in neue Technologien als auch in ihre Belegschaft, indem sie Generationsunterschiede überbrücken und kontinuierliches Lernen und Innovation fördern.

In diesem Leitfaden wird eine schrittweise Migrationsstrategie erörtert, die die Vorteile von Mainframe-Systemen mit den Vorteilen moderner Cloud-Technologien in Einklang bringt. Bei diesem schrittweisen Replatforming-Ansatz wird zunächst die Anwendungsebene migriert und gleichzeitig die Konnektivität zu Ihrer bestehenden IBM Db2 for z/OS Datenbank aufrechterhalten, um den Übergangsprozess zu optimieren und Unterbrechungen Ihrer kritischen Geschäftsabläufe zu minimieren, während Sie neue Funktionen in der Cloud einführen.

Dieser Leitfaden richtet sich an technische Entscheidungsträger und Implementierungsteams, die an Initiativen zur Modernisierung von Mainframes beteiligt sind. Zu den wichtigsten Zielgruppen gehören Unternehmens- und Lösungsarchitekten, technische Projektmanager und Leiter von Modernisierungsprogrammen, die sowohl die strategischen als auch die technischen Aspekte der Mainframe-Neuausrichtung verstehen müssen. Der Inhalt ist ebenso wertvoll für Implementierungsteams wie Mainframe-Anwendungsentwickler AWS oder Cloud-Techniker, Datenbankadministratoren und Techniker, die für die Durchführung und DevOps Implementierung der Modernisierung verantwortlich sind.

Geschäftsergebnisse

Unternehmen haben viele überzeugende Gründe, ihre veralteten Anwendungen zu aktualisieren. Dieser Prozess sorgt in allen Branchen für ein Gefühl der Dringlichkeit. Wenn ältere Experten in den Ruhestand gehen, hinterlassen sie eine erhebliche Wissenslücke, weshalb es entscheidend ist, die Systeme zu modernisieren, bevor dieses Fachwissen verloren geht. Darüber hinaus sind Unternehmen von der Notwendigkeit getrieben, Kosten zu senken, die Agilität zu erhöhen und schnell auf sich schnell ändernde Marktbedingungen zu reagieren.

Das Streben nach digitaler Transformation wird durch neue Technologien und die Nachfrage nach verbesserten Kundenerlebnissen weiter verstärkt. Diese Faktoren, kombiniert mit den Risiken, die mit der Wartung komplexer Systeme verbunden sind, veranlassen Unternehmen, bei der Modernisierung ihrer IT-Infrastruktur schnell zu handeln.

Insbesondere die Modernisierung von Mainframes stellt einen heiklen Balanceakt dar. Unternehmen müssen die Stabilität und Sicherheit wahren, für die Mainframes bekannt sind, und gleichzeitig die Flexibilität und Skalierbarkeit nutzen, die moderne Architekturen bieten. Dieser Prozess beinhaltet komplexe Entscheidungen darüber, welche Anwendungen migriert, welche neu geschrieben und welche auf dem Mainframe beibehalten werden sollen.

Zu den wichtigsten Triebkräften für die Modernisierung gehören Agilität und Kostensenkung:

- **Agilität und Markteinführungszeit.** Moderne Systeme ermöglichen schnellere Beschaffungsprozesse und schnellere Reaktionen auf sich ändernde Marktanforderungen. Die Einführung von DevOps und SysOps Praktiken kann die Produktivität und die Bereitstellungsgeschwindigkeit erheblich verbessern.
- **Kostensenkung.** Modernisierung führt häufig zu niedrigeren Infrastrukturkosten durch:
 - Pay-as-you-go Modelle, die die Kosten an der tatsächlichen Nutzung ausrichten.

- Geringere Lizenzgebühren im Zusammenhang mit älteren Systemen.
- Verbesserte Elastizität, die für eine bessere Ressourcenzuweisung sorgt.
- Active-Active-Setups mit hoher Verfügbarkeit, die die Ausfallsicherheit des Systems erhöhen und gleichzeitig die Ressourcennutzung optimieren.

Auf der Grundlage dieser Geschäftstreiber wird die Neuausrichtung von COBOL-Anwendungen als strategischer Modernisierungsansatz angesehen. Sie können eine gemeinsam genutzte Datenbank verwenden, um einen schrittweisen Migrationspfad zu verfolgen, der den Modernisierungsbedarf mit der Notwendigkeit der Aufrechterhaltung der Geschäftskontinuität in Einklang bringt. Mit dieser Methode können Sie die Vorteile moderner Architekturen nutzen und gleichzeitig die Zuverlässigkeit Ihrer COBOL-Anwendungen wahren. Auf diese Weise können Sie Agilität, Kosteneffizienz und Innovation erreichen und gleichzeitig die Risiken minimieren, die mit groß angelegten, abrupten Übergängen verbunden sind. Der in diesem Leitfaden beschriebene Ansatz für eine gemeinsame DB2-Datenbank bietet eine Brücke zwischen älteren Systemen und modernen Plattformen und ermöglicht einen reibungsloseren, kontrollierteren Modernisierungsprozess.

In diesem Leitfaden:

- [AWS Mainframe Modernization](#)
- [Der Transformationsprozess](#)
- [Best Practices](#)
- [Nächste Schritte](#)
- [Ressourcen](#)
- [Dokumentverlauf](#)

AWS Mainframe Modernization

Note

AWS Mainframe Modernization Der Service (Managed Runtime Environment Experience) steht Neukunden nicht mehr zur Verfügung. Funktionen, die AWS Mainframe Modernization Service (Managed Runtime Environment-Erfahrung) ähneln, finden Sie unter AWS Mainframe Modernization Service (Self-Managed Experience). Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der AWS Mainframe Modernization Verfügbarkeit](#).

Der [AWS Mainframe Modernization Service](#) ermöglicht es Ihnen, Ihre älteren Mainframe-Anwendungen in eine cloudnative Umgebung zu migrieren, bestehende Geschäftslogik und Investitionen beizubehalten, automatisierte Tools und verwaltete Runtime-Services zu verwenden, die Anwendungsleistung zu optimieren und die Betriebskosten zu senken. Dieser Service rationalisiert den Modernisierungsprozess, sodass Sie das Potenzial der Cloud nutzen und gleichzeitig den Wert Ihrer Mainframe-Kernsysteme wahren können. AWS bietet zwei wichtige Ansätze für die Mainframe-Modernisierung: Replatforming und automatisiertes Refactoring.

Plattformwechsel

AWS Mainframe Modernization Die [Replatform mit Rocket Software](#) (ehemals Micro Focus) bietet eine leistungsstarke Replatforming-Option für Unternehmen, die ihre Mainframe-Anwendungen mit minimaler Unterbrechung in die Cloud migrieren möchten. Mit dieser Lösung können Sie Ihre vorhandenen COBOL-Dateien und PL/I Anwendungen neu kompilieren und ausführen, ohne dass wesentliche Codeänderungen erforderlich sind. AWS

Zu den wichtigsten Vorteilen der Lösung AWS Replatform with Rocket Software gehören:

- Beibehaltung der bestehenden Geschäftslogik und Investitionen
- Geringeres Risiko und kürzere Markteinführungszeit
- Verbesserte Skalierbarkeit und Leistung der AWS Infrastruktur
- Zugang zu modernen Entwicklungstools und -praktiken

Mit dieser Lösung können Sie Ihre vertrauten Mainframe-Programmiersprachen beibehalten und gleichzeitig die Flexibilität, Wirtschaftlichkeit und Innovation von nutzen. AWS Cloud

Automatisiertes Refactoring

Für einen transformativeren Ansatz als das Replatforming können Sie [AWS Blu Age](#) verwenden, das automatisiertes Refactoring von Mainframe-Anwendungen auf Java-basierte Cloud-native Anwendungen bietet. Diese Lösung hilft Ihnen dabei, Ihre Altsysteme umfassender zu modernisieren und sie in Anwendungen umzuwandeln, die die Vorteile Cloud-nativer Technologien voll ausschöpfen können.

Zu den wichtigsten Vorteilen von AWS Blu Age gehören:

- Konvertierung von Legacy-Code in moderne, wartbare Java-Anwendungen
- Automatisierte Transformation, die den manuellen Aufwand und potenzielle Fehler reduziert
- Erstellung von Cloud-nativen Anwendungen, die optimiert sind für AWS-Services
- Verbesserte Agilität und einfachere Integration mit modernen Technologien

AWS Blu Age hilft Ihnen dabei, Ihre Anwendungen zu migrieren und sie für die Cloud vorzubereiten, um neue Möglichkeiten für Innovation und Wachstum zu eröffnen. Weitere Informationen zu diesem Ansatz finden Sie in der Dokumentation unter [Automatisches Refactoring von Anwendungen mit AWS Blu Age](#). AWS Mainframe Modernization

Vorteile von Replatforming

In diesem Handbuch wird ein Ansatz für die Neuplattformierung von Mainframe-COBOL-Anwendungen auf beschrieben. AWS Dieser Ansatz zielt darauf ab, ältere Systeme zu modernisieren und gleichzeitig IBM Db2 vorübergehend beizubehalten, um den Übergangsprozess zu z/OS rationalisieren. Indem Sie zunächst die bestehende Datenbankstruktur beibehalten, können Sie die Komplexität und das Risiko bei der Migration reduzieren. Dieser schrittweise Ansatz hilft Ihnen, von der Skalierbarkeit und Kosteneffektivität von zu profitieren und AWS Cloud gleichzeitig die Integrität kritischer Daten zu wahren. Zu den Vorteilen einer schrittweisen Neuplattformierung gehören:

- Beschleunigte Modernisierung: Replatforming und Refactoring erfordern in der Regel weniger Zeit und Ressourcen als die Neugestaltung einer Legacy-Anwendung in der Cloud, da nicht die gesamte Anwendung neu geschrieben werden muss. Dieser Ansatz unterstützt auch

einen schrittweisen Übergang, der es Unternehmen ermöglicht, in ihrem eigenen Tempo zu modernisieren und gleichzeitig sofort von der Skalierbarkeit und Kosteneffektivität der zu profitieren. AWS Cloud

- **Risikominderung:** Replatforming bietet für viele Unternehmen mehrere Vorteile gegenüber Refactoring. Unternehmen können ihre bestehenden COBOL- und PL/I Codebasen beibehalten, jahrelange Geschäftslogik beibehalten und das mit umfangreichen Codeänderungen verbundene Risiko minimieren.
- **Datenkontinuität und schrittweise Migration:** Ein wesentlicher Vorteil der Neuausrichtung ist die Option, Daten zunächst in Db2 für in ihrem ursprünglichen Datenformat beizubehalten. z/OS Diese Strategie vermeidet die Notwendigkeit sofortiger, komplexer und potenziell riskanter Datenmigrationsprozesse. Indem Sie die Daten in der Anfangsphase in ihrer ursprünglichen Umgebung beibehalten, können Sie die Datenintegrität wahren, Ausfallzeiten reduzieren und das Risiko von Datenverlust oder Datenbeschädigung während des Modernisierungsprozesses minimieren. Als zweiten Schritt können Sie eine kontrollierte, schrittweise Datenmigration zu cloudnativen Datenbanken planen, die gründliche Tests und Validierungen umfasst, während die Anwendung weiterhin in der Umgebung mit der neuen Plattform ausgeführt wird.
- **Flexibilität und Zukunftssicherheit:** Für Unternehmen, die erheblich in Mainframe-Fähigkeiten und -Anwendungen investieren, bietet Replatforming einen pragmatischen Weg zur Modernisierung, der Innovation und Kontinuität in Einklang bringt. Es bietet die Flexibilität, kritische Datenstrukturen und Zugriffsmethoden zunächst beizubehalten und schafft gleichzeitig die Voraussetzungen für future Modernisierungsmaßnahmen, einschließlich einer eventuellen Datenmigration zu vollständig cloudnativen Lösungen.

Organizations können den Replattform-Ansatz verfolgen, um in ihrem eigenen Tempo zu modernisieren und auf unmittelbare Bedürfnisse einzugehen und gleichzeitig langfristige Ziele für die digitale Transformation zu planen. Dieser Ansatz bietet Unternehmen auch die Möglichkeit, ihre Mitarbeiter in Cloud-nativen Diensten zu schulen.

Der Transformationsprozess

Die Mainframe-Modernisierung ist ein entscheidender Schritt für Unternehmen, die die Vorteile des Cloud-Computing nutzen und gleichzeitig ihre wertvollen Legacy-Anwendungen erhalten möchten. Diese Transformation bringt erhebliche Herausforderungen mit sich. Mainframe-Anwendungen sind in der Regel stark miteinander verknüpft und weisen komplizierte Interdependenzen auf, die sich im Laufe jahrzehntelanger Betriebszeit entwickelt haben. Diese Komplexität erfordert einen sorgfältigen und methodischen Modernisierungsansatz.

Für eine erfolgreiche Umstellung müssen Organizations die folgenden Schlüsselphasen bewältigen:

- **Planung:** Diese Phase beinhaltet eine umfassende Untersuchung der vorhandenen Systeme und die Priorisierung der Modernisierungsmaßnahmen. Organizations bewerten ihre aktuelle Infrastruktur, identifizieren kritische Anwendungen und bestimmen, welche Systeme zuerst modernisiert werden müssen.
- **Aufbau:** In dieser Phase entwickeln Unternehmen Prozesse zur Migration von Anwendungen und zur Entwicklung neuer Systeme und Infrastrukturen. Dies beinhaltet den Entwurf und die Implementierung der modernisierten Architektur sowie die Kompilierung des Quellcodes.
- **Ausführen:** Dieser Schritt besteht aus der Erstellung der Laufzeitumgebungen zum Hosten der Anwendungen auf der neuen Plattform. Dazu gehört die Einrichtung der erforderlichen Hardware-, Software- und Cloud-Infrastruktur, um die modernisierten Systeme zu unterstützen und sicherzustellen, dass sie in der neuen Umgebung effizient arbeiten können.
- **Testen:** Diese Phase umfasst eine strenge Validierung der modernisierten Systeme, um sicherzustellen, dass alle Funktions- und Leistungsanforderungen erfüllt wurden. Umfangreiche Tests werden durchgeführt, um die Datenintegrität, Systemkompatibilität und Gesamtleistung der neuen Umgebung zu überprüfen.
- **Umstellung:** Die letzte Phase konzentriert sich auf die Implementierung von Strategien für einen reibungslosen Übergang und die Steuerung der Umstellung vom alten Mainframe auf die modernisierte Umgebung. Dies beinhaltet eine sorgfältige Planung des Migrationsplans und der Notfallpläne, um Störungen des Geschäftsbetriebs so gering wie möglich zu halten.

In den folgenden Abschnitten werden diese Phasen ausführlich behandelt:

- [Planung](#)
- [Gebäude](#)

- [In Ausführung](#)
- [Testen](#)
- [Umstellung](#)

Planung

Um die Anforderungen älterer Mainframe-Anwendungen effektiv zu bewältigen, beginnen Unternehmen häufig mit einer umfassenden Bewertung ihrer Mainframe-Umgebung.

Entdeckung von Anwendungen

Ein leistungsstarkes Tool in dieser Anfangsphase ist der [Rocket Enterprise Analyzer](#), der tiefe Einblicke in die Struktur, Abhängigkeiten und Komplexität von Mainframe-Anwendungen bietet. Dieses Tool hilft Ihnen dabei, den Umfang Ihrer Modernisierungsmaßnahmen, potenzielle Risiken und Optimierungsmöglichkeiten zu ermitteln.

Ein entscheidender Aspekt, den es aufzudecken gilt, ist das komplizierte Geflecht von Datenabhängigkeiten innerhalb von Mainframe-Systemen. Diese Abhängigkeiten sind oft unter Schichten von veraltetem Code verborgen und können sich erheblich auf Modernisierungsbemühungen auswirken. Indem Sie aufzeigen, wie verschiedene Anwendungen und Module mit verschiedenen Datenquellen interagieren, können Sie die potenziellen Auswirkungen aller Änderungen, die Sie implementieren möchten, besser verstehen.

Abhängigkeiten von Daten

Eine gründliche Bewertung der Datenabhängigkeiten kann wichtige Informationen über Datenfluss, Datenqualität und Datenverwaltung in Ihrer Mainframe-Umgebung liefern. Dieses Wissen ist von unschätzbarem Wert bei der Planung von Datenmigrationsstrategien, der Sicherstellung der Datenintegrität bei der Modernisierung und der Identifizierung von Möglichkeiten zur Datenoptimierung. Wenn Sie sich ein klares Bild von Ihren Daten machen, können Sie fundiertere Entscheidungen darüber treffen, welche Modernisierungsansätze am effektivsten sind und Ihre bestehenden Abläufe am wenigsten stören.

Eine Top-down-Analyse, die die Verwendung von Tabellen durch Transaktionen oder Job Control Language (JCL) -Jobs identifiziert, ist entscheidend für die Planung und Priorisierung von Wellen. Dieser Ansatz verdeutlicht die Beziehungen zwischen den verschiedenen Komponenten Ihrer

Mainframe-Systeme und unterstützt Sie bei der Entwicklung eines strategischen, schrittweisen Modernisierungsansatzes. Indem Sie ermitteln, auf welche Tabellen am häufigsten zugegriffen wird und über welche Prozesse, können Sie Ihre Modernisierungsmaßnahmen priorisieren: Sie können sich zunächst auf Bereiche mit großer Wirkung konzentrieren und einen reibungsloseren Übergang mit minimaler Unterbrechung kritischer Geschäftsabläufe sicherstellen.

Neben der Nutzung von Rocket Enterprise Analyzer zur Erkennung von Datenabhängigkeiten nutzen viele Unternehmen auch ihre eigenen maßgeschneiderten Lösungen, um tiefere Einblicke in ihre Mainframe-Umgebungen zu gewinnen. Diese internen Tools nutzen häufig die Fülle an Informationen, die im IBM Db2-Katalog und in den SMF-Datensätzen (System Management Facility) verfügbar sind.

Kapazitäts-Benchmark

Ein Schritt bei der Planung Ihres Mainframe-Replattforming-Projekts besteht darin, detaillierte Informationen über Ihren aktuellen Workload-Verbrauch zu sammeln. Diese Daten helfen Ihnen dabei, die anfänglich benötigte Kapazität in Ihrer Ziel-Cloud-Umgebung genau vorherzusagen und bereitzustellen. Wir empfehlen beispielsweise, dass Sie Daten zum stündlichen Verbrauch von Millionen Instruktionen pro Sekunde (MIPS) sowohl für Online-Transaktionen als auch für Batch-Transaktionen aus IBM Customer Information Control System (CICS) oder Information Management System (IMS) und Job Control Language (JCL) -Jobs sammeln.

IBM bietet eine Vielzahl von [Preismodellen](#) für MIPS im Mainframe-Computing an, und viele dieser Modelle konzentrieren sich auf die Spitzennutzung. Unter diesen Modellen, die auf Spitzenzeiten basieren, ist der rollierende Spitzenwert von vier Stunden am gebräuchlichsten.

Die Mainframe-Kosten umfassen fünf Hauptbereiche, die sich erheblich auf die Gesamtkosten auswirken:

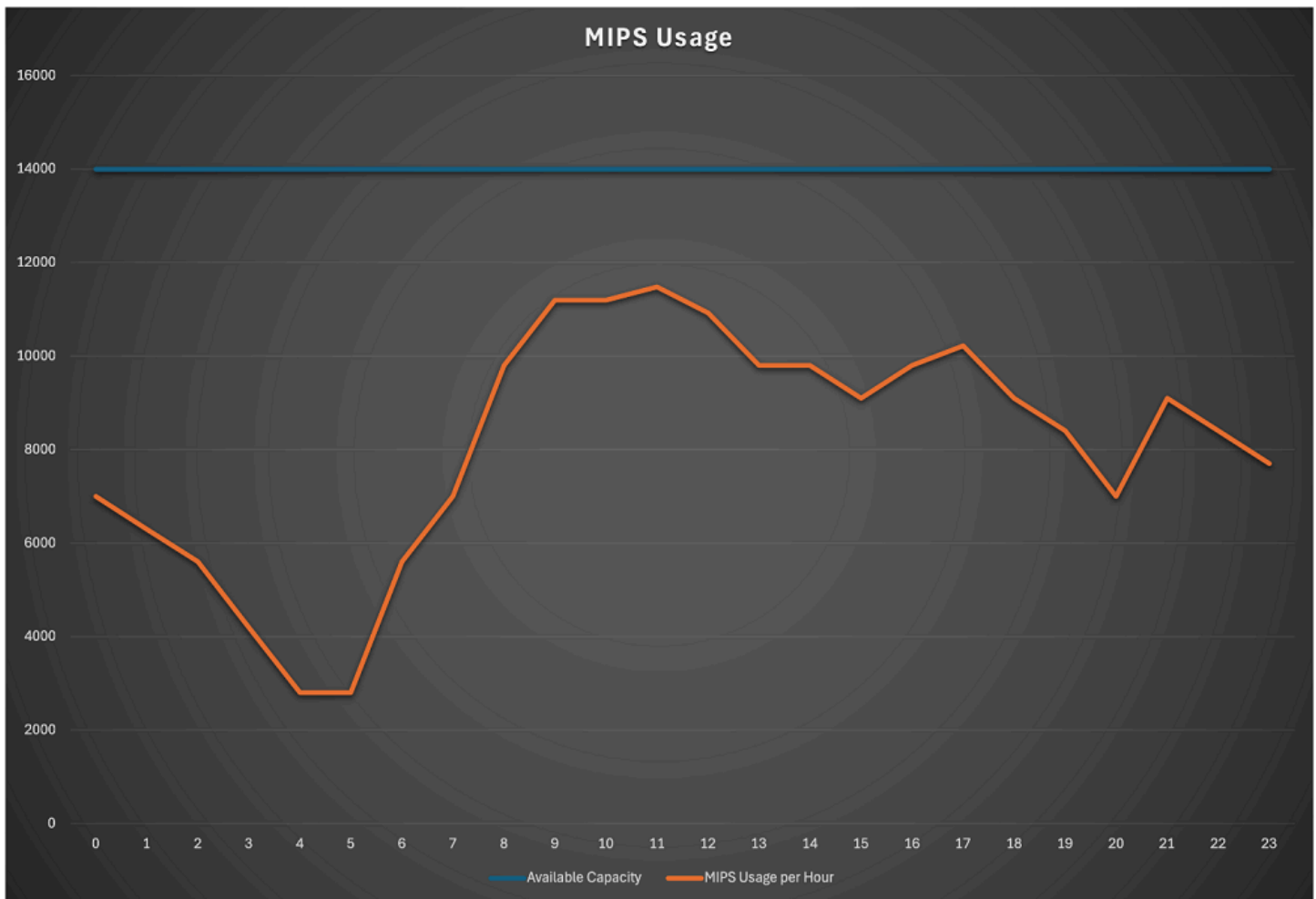
- Die Softwarelizenzierung ist oft ein wichtiger Bestandteil. Sie deckt Betriebssysteme, Middleware, Datenbanken und verschiedene Anwendungen ab, und die Kosten hängen manchmal von der Maschinenkapazität oder -nutzung ab.
- Zu den Hardwarekosten gehören der anfängliche Kauf oder das Leasing von Mainframe-Geräten, die laufende Wartung und Upgrades.
- Die Speicherkosten können aufgrund der riesigen Menge verwalteter Daten, die verwaltet werden, erheblich sein. Dazu gehören Festplattensysteme, Bandbibliotheken und zugehörige Verwaltungssoftware.

-
- Die Personalkosten decken die Gehälter spezialisierter Mainframe-Experten wie Systemprogrammierer und Datenbankadministratoren.
 - Maßnahmen zur Notfallwiederherstellung und Geschäftskontinuität, darunter Backup-Systeme, redundante Hardware und externe Wiederherstellungseinrichtungen, stellen eine erhebliche Investition dar, um hohe Verfügbarkeit und schnelle Wiederherstellung zu gewährleisten.

Diese fünf Kostenkategorien bilden zusammen mit MIPS-Gebühren den Kern der meisten Mainframe-Budgets. Ihre relativen Anteile können jedoch je nach Größe, Branche und spezifischen Mainframe-Nutzungsmustern Ihres Unternehmens stark variieren.

Stündliche MIPS-Daten sind entscheidend für ein umfassendes Verständnis der Workload-Muster und der Leistung Ihres Mainframes. Im Gegensatz zu Tages- oder Monatsdurchschnittswerten bieten stündliche Daten detaillierte Einblicke, die die nuancierten Schwankungen der Ressourcennutzung Ihres Systems im Laufe des Tages aufdecken. Dieser Detaillierungsgrad ist von unschätzbarem Wert für die genaue Bewertung der Leistungs- und Kapazitätsanforderungen Ihrer Anwendung in der Cloud.

Durch die Analyse stündlicher MIPS-Daten können Sie Spitzennutzungsperioden identifizieren, Trends erkennen und potenzielle Engpässe lokalisieren, die in aggregierten Daten möglicherweise verdeckt sind, wie in der folgenden Abbildung dargestellt. Diese Granularität ermöglicht eine genauere Kapazitätsplanung, trägt zur Optimierung der Ressourcenzuweisung bei und kann potenziell zu Kosteneinsparungen und einer verbesserten Systemeffizienz führen.



Stündliche MIPS-Daten dienen auch als unverzichtbares Tool für Leistungsbenchmarks. Sie bilden einen detaillierten Basiswert für die Leistung Ihres Systems, was besonders nützlich ist, wenn Sie Systemänderungen wie Migrationen oder Upgrades planen oder evaluieren. Durch den Vergleich stündlicher MIPS-Daten vor und nach der Änderung können Sie die Auswirkungen dieser Änderungen auf die Leistung Ihres Systems genau messen und sicherstellen, dass Ihr Mainframe weiterhin den Anforderungen Ihres Unternehmens entspricht.

Um stündliche MIPS-Daten zu sammeln, haben Sie mehrere Möglichkeiten. Ein Ansatz besteht darin, SMF-Datensätze direkt zu verwenden. Diese Aufzeichnungen bieten eine Fülle von Informationen über Systemaktivitäten und Ressourcennutzung. Alternativ können Sie spezielle Tools wie das IBM Sub-Capacity Reporting Tool (SCRT) verwenden, die den Prozess der Erfassung und Analyse von MIPS-Daten vereinfachen können.

Unabhängig von der Methode, die Sie wählen, ist es wichtig, Daten über einen längeren Zeitraum zu sammeln — idealerweise über mehrere Monate. Dieser erweiterte Erfassungszeitraum ermöglicht

es Ihnen, zyklische Schwankungen Ihrer Arbeitslast zu berücksichtigen, z. B. end-of-month Verarbeitungsspitzen oder saisonale Schwankungen. Durch die Erfassung dieser langfristigen Muster können Sie sich ein genaueres und umfassenderes Bild der Leistungsmerkmale Ihres Mainframes machen, was fundiertere Entscheidungen und ein effektiveres Kapazitätsmanagement ermöglicht.

Wellenplanung

Sie können die gesammelten Informationen nutzen, um Ihre Mainframe-Replattforming-Initiativen strategisch zu priorisieren. Ein umsichtiger Ansatz besteht darin, mit weniger kritischen Workloads zu beginnen, wie z. B. Transaktionen, die nicht zum Kerngeschäft gehören, oder Batch-Jobs, damit die Teams Erfahrungen sammeln und Prozesse verfeinern können, ohne dass das Risiko für wichtige Abläufe minimiert wird. Darüber hinaus kann es von Vorteil sein, schreibgeschützte Workloads als frühe Kandidaten für eine Migration in Betracht zu ziehen, da diese Workloads in der Regel weniger komplex und mit einem geringeren Risiko von Dateninkonsistenzen verbunden sind. Dieser Ansatz ermöglicht es Ihnen, Vertrauen und Dynamik bei Ihren Umstellungsbemühungen aufzubauen.

Darüber hinaus kann die Gruppierung von Workloads, die sich Db2-Tabellen für Schreib- oder Aktualisierungsvorgänge teilen, den Migrationsprozess rationalisieren. Durch die Identifizierung dieser miteinander verbundenen Workloads können Sie zusammenhängende Migrationswellen planen, die die Datenintegrität wahren und den Bedarf an komplexen Zwischenlösungen minimieren. Diese Strategie reduziert nicht nur das Risiko von Datenkonflikten, sondern optimiert auch den Gesamtzeitplan für die Umstellung der Plattform, indem verwandte Komponenten gleichzeitig berücksichtigt werden. Letztlich sorgt dieser datengesteuerte Priorisierungsansatz für eine ausgewogene Berücksichtigung von Kritikalität, Komplexität und Interdependenz und führt zu einem effizienteren und erfolgreicherem Modernisierungsprozess für Mainframes.

Erstellung

Die Verwendung einer gemeinsam genutzten Db2-Datenbank ermöglicht die gleichzeitige Ausführung identischer oder konsistenter Anwendungen sowohl in Mainframe- als auch in Cloud-Umgebungen. Dieser Ansatz bietet mehrere Vorteile, wenn Sie auf beiden Plattformen dieselbe Anwendungsversion verwenden, und sorgt für mehr Flexibilität und Zuverlässigkeit in Ihrem Betrieb.

Ein entscheidender Vorteil dieser Strategie ist die Möglichkeit, einen effektiven Rollback-Plan zu implementieren. Wenn während der Migration oder Bereitstellung Probleme auftreten, ermöglicht die Verwendung derselben Anwendungsversion eine nahtlose Rückkehr zum vorherigen Status und minimiert Ausfallzeiten und potenzielle Dateninkonsistenzen.

Konsistenz der Anwendungen

Die Spiegelung von Anwendungskomponenten von einem Distributed Source Control Manager auf den Mainframe ist ein strategischer Ansatz während des Umstellungsprozesses. Diese Methode unterstützt die Verwendung moderner Tools zur Quellcodeverwaltung und gewährleistet gleichzeitig die Synchronisation mit der Mainframe-Umgebung. Dieser Spiegelungsprozess ist temporär und dauert nur so lange, bis der Workload in der Produktion auf der verteilten Plattform voll funktionsfähig ist.

Durch die Migration des Quellcodes Ihrer Anwendung auf eine neue Plattform zu einem verteilten Change-Management-Tool können Sie mehrere Vorteile nutzen, die moderne Quellcode-Manager bieten. Dazu zählen:

- **Verbesserte Zusammenarbeit:** Verteilte Tools bieten häufig eine bessere Unterstützung für die Teamzusammenarbeit, indem sie Funktionen wie Pull-Requests, Codeüberprüfungen und Branching-Strategien enthalten.
- **Verbesserte Versionskontrolle:** Moderne Systeme bieten eine detailliertere Versionskontrolle und erleichtern die Nachverfolgung von Änderungen und die Verwaltung verschiedener Versionen des Codes.
- **Integration mit CI/CD Pipelines:** Viele verteilte Tools lassen sich nahtlos in CI/CD-Pipelines (Continuous Integration and Continuous Deployment) integrieren, wodurch der Entwicklungsprozess optimiert wird.
- **Bessere Transparenz und Rückverfolgbarkeit:** Diese Tools bieten häufig überlegene Dashboards und Berichtsfunktionen und bieten einen besseren Einblick in den Entwicklungsprozess.
- **Support moderner Entwicklungspraktiken:** Verteilte Systeme eignen sich in der Regel besser für agile Methoden und DevOps Praktiken.

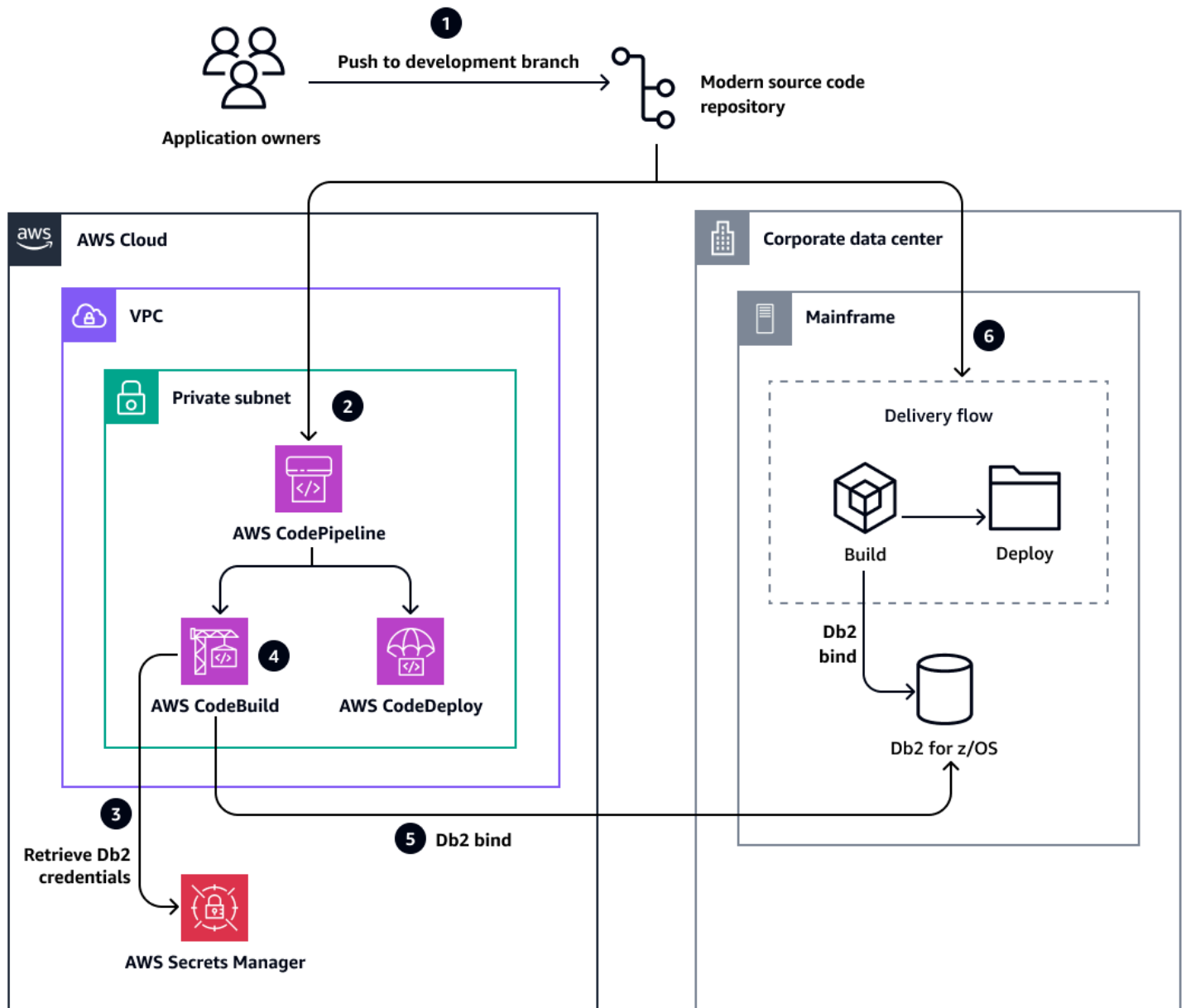
Bei der Spiegelung wird der Code vom Distributed Source Control Manager zurück zum Mainframe synchronisiert. Dadurch wird sichergestellt, dass beide Umgebungen während der Übergangsphase konsistent bleiben. Sie müssen die Spiegelung jedoch als unidirektionale Synchronisation implementieren, bei der Updates nicht bidirektional, sondern vom verteilten System zum Mainframe fließen. Dieser Ansatz gewährleistet die Konsistenz und verhindert potenzielle Konflikte, die durch gleichzeitige Updates in beiden Umgebungen entstehen könnten.

Mit dieser Spiegelungsstrategie können Sie Ihre Entwicklungsaktivitäten schrittweise auf die verteilte Plattform verlagern und gleichzeitig sicherstellen, dass die Mainframe-Umgebung erhalten

bleibt. up-to-date Dies sorgt für einen reibungsloseren Übergang und ein Sicherheitsnetz während des Umstellungsprozesses. Wenn der Workload in der verteilten Produktionsumgebung voll funktionsfähig und stabil ist, können Sie den Spiegelungsprozess schrittweise beenden und die Migration zum modernen Quellcodeverwaltungssystem abschließen.

Architektur

Das folgende Diagramm zeigt, wie ein verteiltes Quellcodeverwaltungssystem Anwendungskomponenten spiegeln und die Synchronisation zwischen der Umgebung AWS Cloud und der Mainframe-Umgebung aufrechterhalten kann. Die AWS Cloud Umgebung verwendet CI/CD Dienste wie [AWS CodeBuild](#), und [AWS CodePipeline](#), um die Anwendung [AWS CodeDeploy](#) zu erstellen und bereitzustellen.



In diesem Workflow:

1. Die Anwendungseigentümer stellen eine neue Anwendungsversion in den Entwicklungszweig des Quellcode-Repositorys bereit.
2. Die neue Version wird ausgelöst AWS CodePipeline.
3. AWS CodeBuild ruft Db2-Anmeldeinformationen von ab. [AWS Secrets Manager](#)
4. CodeBuild kompiliert die Anwendung.
5. CodeBuild verwendet Db2 for z/OS , um die Anwendung zu binden.
6. Der Mainframe-Bereitstellungsablauf erstellt und implementiert die Anwendung ebenfalls.

In Ausführung

Um eine optimale Leistung und geringe Latenz zwischen Ihrer cloudbasierten Anwendung und Ihrer lokalen Datenbank zu gewährleisten, empfehlen wir die Implementierung. [AWS Direct Connect](#) Dieser Service bietet eine dedizierte Netzwerkverbindung zwischen AWS und dem Rechenzentrum Ihres Unternehmens und bietet im Vergleich zu internetbasierten Verbindungen eine konsistentere Netzwerkleistung und eine geringere Latenz. Dies ist besonders wichtig für Datenbankoperationen, die schnelle Reaktionszeiten erfordern.

Um eine hohe Verfügbarkeit (HA) und Elastizität der Anwendung zu erreichen, auf der sie ausgeführt wird AWS, können Sie mithilfe der folgenden Komponenten eine robuste Architektur implementieren:

- Elastic Load Balancing (ELB): Sie können einen Load Balancer einsetzen, um eingehenden Datenverkehr auf mehrere Amazon Elastic Compute Cloud (Amazon EC2) -Instances zu verteilen, auf denen Ihre Anwendung ausgeführt wird. Dies gewährleistet eine gleichmäßige Verteilung der Arbeitslast und bietet einen zentralen Einstiegspunkt für Kundenanfragen.
- Auto Scaling Scaling-Gruppe: EC2-Instances, die die Anwendung hosten, können in einer Auto Scaling Scaling-Gruppe organisiert werden. Auf diese Weise kann die Infrastruktur die Anzahl der Instances auf der Grundlage vordefinierter Kennzahlen wie CPU-Auslastung oder Netzwerkverkehr automatisch anpassen. In Spitzenzeiten können zusätzliche Instances gestartet werden, um die erhöhte Last zu bewältigen, wohingegen in ruhigeren Zeiten nicht benötigte Instanzen beendet werden können, um die Kosten zu optimieren.
- EC2-Instances: Die Anwendung kann auf EC2-Instances innerhalb der Auto Scaling Scaling-Gruppe bereitgestellt werden. Diese Instances sollten auf mehrere Availability Zones verteilt werden, um die Fehlertoleranz zu verbessern und eine hohe Verfügbarkeit sicherzustellen.
- Multi-AZ-Bereitstellung: Durch die Verteilung der Anwendungsinstanzen auf mehrere Availability Zones kann das System den Ausfall einer einzelnen Availability Zone ohne nennenswerte Auswirkungen auf die Gesamtverfügbarkeit überstehen.

Diese Architektur ermöglicht es der Anwendung, je nach Bedarf nahtlos zu skalieren und gleichzeitig die hohe Verfügbarkeit aufrechtzuerhalten. Der Load Balancer stellt sicher, dass der Datenverkehr gleichmäßig auf fehlerfreie Instances verteilt wird, und die Auto Scaling Scaling-Gruppe verwaltet die Anzahl der Instances auf der Grundlage der tatsächlichen Arbeitslast.

Um die Zuverlässigkeit weiter zu erhöhen, können Sie ein robustes Überwachungs- und Warnsystem implementieren, indem Sie [Amazon](#) verwenden CloudWatch, um Leistungsprobleme oder Ausfälle

umgehend zu erkennen und darauf zu reagieren. Darüber hinaus wird durch regelmäßige Tests der automatischen Skalierungsfunktionen und der Failover-Szenarien sichergestellt, dass sich das System bei verschiedenen Lastzuständen und potenziellen Ausfällen erwartungsgemäß verhält.

Mit diesem Ansatz können Sie von der Skalierbarkeit und Flexibilität des profitieren und AWS Cloud gleichzeitig eine sichere Verbindung zu Ihrer lokalen Db2-Datenbank aufrechterhalten. Dieses Hybrid-Setup ist ein hervorragender Weg zu einer vollständigen Cloud-Migration und ermöglicht einen schrittweisen Übergang und Risikominderung während des gesamten Prozesses.

Zweiphasiges Commit (2 PC)

[AWS Mainframe Modernization Replatform with Rocket Software](#) bietet durch die Implementierung einer erweiterten Architektur (XA) Unterstützung für zweiphasige Commit-Transaktionen (2PC). Diese Fähigkeit ist entscheidend für die Aufrechterhaltung der Datenintegrität in verteilten Systemen, insbesondere in Mainframe-Umgebungen, in denen komplexe Transaktionen oft mehrere Ressourcen umfassen.

Die XA-Architektur, die mit Rocket Software in AWS Replatform integriert ist, ermöglicht die Koordination von Transaktionen über verschiedene Ressourcen wie Datenbanken und Nachrichtenwarteschlangen hinweg. Diese Integration stellt sicher, dass alle Teile einer verteilten Transaktion entweder gleichzeitig festgeschrieben oder rückgängig gemacht werden, um die Konsistenz im gesamten System aufrechtzuerhalten.

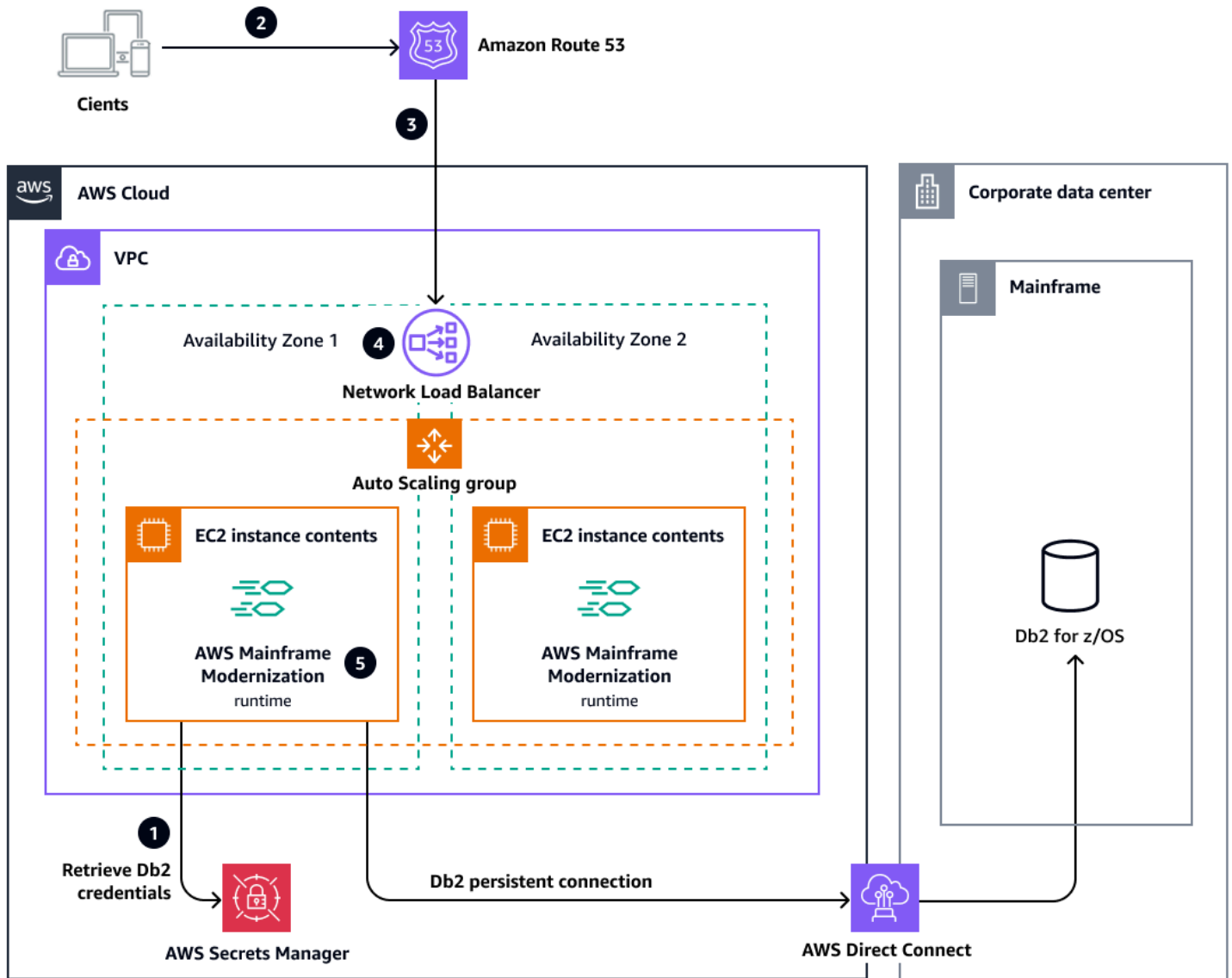
Der 2PC-Prozess besteht aus zwei Phasen:

- Vorbereitungsphase: Der Transaktionsmanager fragt alle an der Transaktion beteiligten Ressourcenmanager ab, um sicherzustellen, dass sie bereit sind, einen Commit durchzuführen.
- Commit-Phase: Wenn alle Ressourcenmanager positiv reagieren, weist der Transaktionsmanager sie an, die Änderungen zu übernehmen. Wenn einer der Ressourcenmanager die Änderungen nicht bestätigen kann, werden alle Manager angewiesen, die Änderungen rückgängig zu machen.

Durch den Einsatz von XA bietet AWS Replatform with Rocket Software eine zuverlässige und skalierbare Lösung für die Verwaltung komplexer, verteilter Transaktionen in modernisierten Mainframe-Umgebungen. Diese Funktion ist unverzichtbar für Unternehmen, die ihre Mainframe-Anwendungen in die Cloud migrieren möchten, ohne Kompromisse bei der Transaktionsintegrität oder Leistung einzugehen.

Laufzeit-Infrastruktur

Das folgende Diagramm zeigt eine hochverfügbare und elastische Umgebung in der, AWS Cloud die zwei Availability Zones, EC2-Instances in einer Auto Scaling Scaling-Gruppe, einen Network Load Balancer und eine dedizierte Verbindung zwischen der AWS und Mainframe-Umgebungen umfasst. AWS Direct Connect



In dieser Architektur:

1. Wenn die AWS Mainframe Modernization Runtime gestartet wird, ruft sie die Db2-Anmeldeinformationen von Db2 for z/OS ab [AWS Secrets Manager](#) und öffnet eine dauerhafte Verbindung mit Db2.

Note

AWS Mainframe Modernization Der Service (Managed Runtime Environment Experience) steht neuen Kunden nicht mehr zur Verfügung. Funktionen, die AWS Mainframe Modernization Service (Managed Runtime Environment-Erfahrung) ähneln, finden Sie unter AWS Mainframe Modernization Service (Self-Managed Experience). Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der AWS Mainframe Modernization Verfügbarkeit](#).

2. Clients binden die Network Load Balancer Balancer-Adresse in [Amazon Route 53](#).
3. Route 53 leitet Transaktionen an den Network Load Balancer weiter.
4. Der Network Load Balancer verteilt Transaktionen auf mehrere EC2-Instances.
5. Der Workload, auf dem ausgeführt wird, AWS Mainframe Modernization interagiert mit Db2 for, z/OS indem er eine persistente Verbindung über AWS Direct Connect

Testen

Wenn Sie eine COBOL-Anwendung auf eine neue Plattform umstellen und gleichzeitig Db2 z/OS als gemeinsam genutzte Datenbank beibehalten, müssen Sie unbedingt sicherstellen, dass das neue System genauso funktioniert wie das Original. Diese hybride Umgebung bietet einzigartige Herausforderungen und Testmöglichkeiten. Die folgende Strategie beschreibt einen umfassenden Ansatz für funktionale Äquivalenztests und dient der Validierung der Leistung, Datenintegrität und nahtlosen Integration der Anwendung mit der vorhandenen Datenbank Db2 for z/OS.

Identifizieren Sie zunächst die kritischen Geschäftsprozesse und Transaktionen, die zwischen den Systemen verglichen werden müssen. Erstellen Sie anschließend einen detaillierten Testplan mit spezifischen Szenarien, mit denen die funktionale Gleichwertigkeit dieser Transaktionen effektiv bewertet werden kann. Entwickeln Sie abschließend umfassende Testdatensätze, die alle identifizierten Szenarien abdecken, und stellen Sie sicher, dass sie für beide Systeme identisch sind, um einen genauen Vergleich zu ermöglichen.

Quellumgebung

- Erster Snapshot (erster Snapshot):

- Stellen Sie sicher, dass die Datentabelle während des Tests nicht von anderen Anwendungen verwendet wird, da dies den Äquivalenztest beeinflussen kann.
- Erstellen Sie einen Snapshot von Db2 für z/OS Tabellen, die von der Transaktion verwendet werden, bevor Sie Tests ausführen.
- Testen des Quellsystems:
 - Führen Sie die gesamte Testsuite auf der ursprünglichen COBOL-Anwendung aus.
 - Zeichnen Sie alle Transaktionen, Eingaben und Ausgaben auf.
 - Überwachen Sie die Systemleistung und die Ressourcennutzung.
- Snapshot nach dem Testen (zweiter Snapshot):
 - Erstellen Sie nach Abschluss der Quellsystemtests einen weiteren Snapshot der z/OS Datenbank Db2 for Database.

Zielumgebung

- Datenbank zurückgesetzt:
 - Stellen Sie den ursprünglichen Zustand der Datenbank mithilfe des ersten Snapshots wieder her.
- Testen des Zielsystems (Umgebung mit neuer Plattform):
 - Führen Sie dieselbe Testsuite auf der neuen Plattform aus.
 - Stellen Sie sicher, dass alle Zielsystemtests dieselben Eingaben verwenden wie die Quellsystemtests.
 - Überwachen Sie die Systemleistung und die Ressourcennutzung.
- Snapshot nach dem Testen des Ziels (dritter Snapshot):
 - Erstellen Sie nach Abschluss der Zielsystemtests einen letzten Snapshot der z/OS Datenbank Db2 for Database.

Analyse

- Vergleich und Analyse:
 - Vergleichen Sie den zweiten und dritten Schnappschuss, um etwaige Datenunstimmigkeiten zu ermitteln.
 - Analysieren Sie die Testergebnisse und vergleichen Sie die Ergebnisse der Quell- und Zielsysteme.

- Bewerten Sie die Leistungskennzahlen zwischen den beiden Umgebungen.
- Integrationstests:
 - Führen Sie Tests durch, die sowohl die Anwendung auf die neue Plattform als auch alle verbleibenden COBOL-Komponenten einbeziehen.
 - Stellen Sie die reibungslose Interaktion zwischen den beiden Umgebungen sicher.
- Failover- und Wiederherstellungstests:
 - Testen Sie Szenarien, in denen eine Umgebung ausfällt und die andere Umgebung die Kontrolle übernimmt.
 - Stellen Sie Datenkonsistenz und Integrität in Failover-Situationen sicher.
- Last- und Stresstests:
 - Führen Sie Tests mit unterschiedlichen Belastungen durch, um zu beurteilen, wie sich das Hybridsystem unter stress verhält.
 - Identifizieren Sie alle Engpässe oder Leistungsprobleme in beiden Umgebungen.
- Dokumentation und Berichterstattung:
 - Dokumentieren Sie alle Testergebnisse, Abweichungen und Leistungskennzahlen.
 - Erstellen Sie einen umfassenden Bericht, in dem Quell- und Zielsystem verglichen werden.

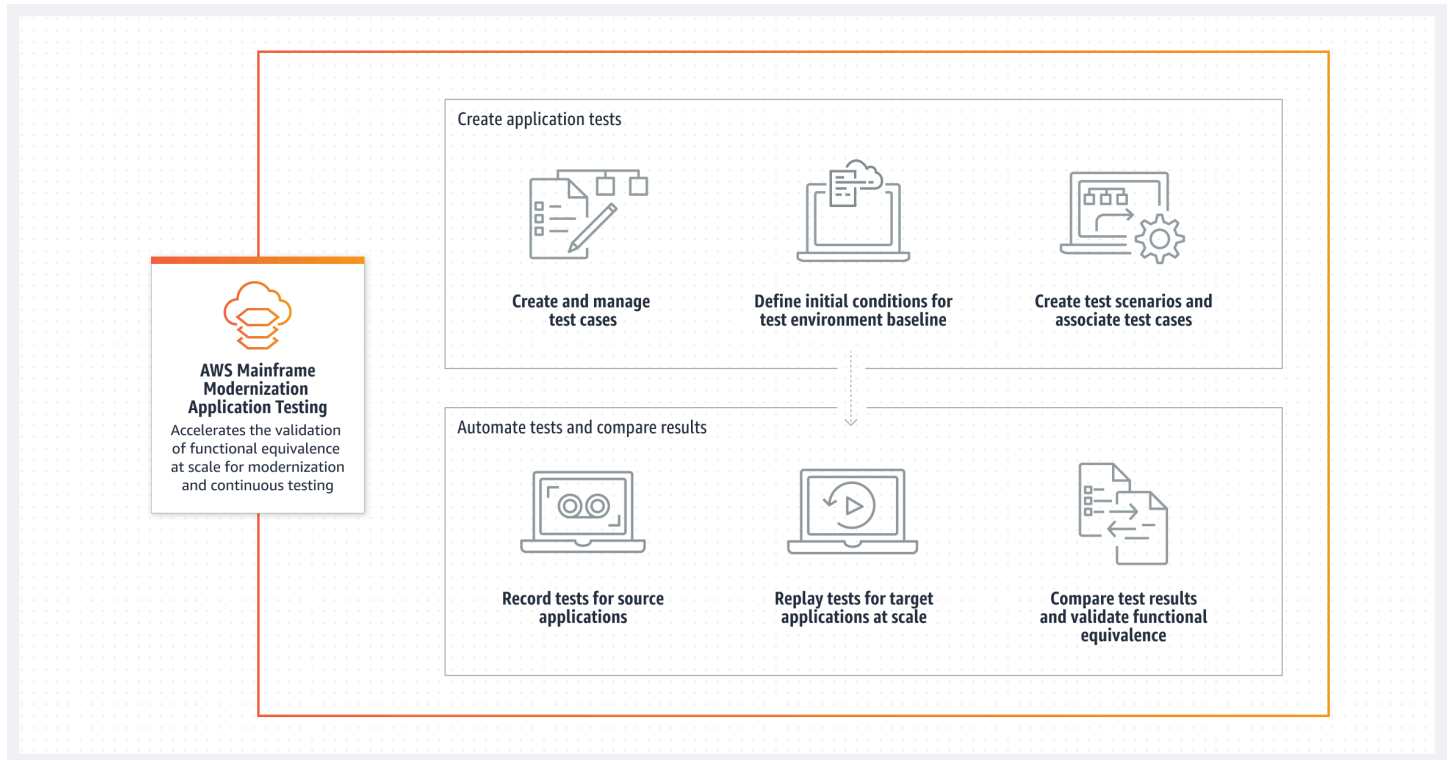
Testen Sie Ihre Anwendung in AWS Mainframe Modernization

Der [Anwendungstests für AWS Mainframe Modernization](#) Service automatisiert die Ausführung von Anwendungstests in großem Maßstab. Anwendungstests für AWS trägt dazu bei, die Projektkosten für die Modernisierung und das Testen von Mainframe-Anwendungen zu optimieren und zu senken.

Note

AWS Mainframe Modernization Der Service (Managed Runtime Environment Experience) steht neuen Kunden nicht mehr zur Verfügung. Funktionen, die AWS Mainframe Modernization Service (Managed Runtime Environment-Erfahrung) ähneln, finden Sie unter AWS Mainframe Modernization Service (Self-Managed Experience). Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der AWS Mainframe Modernization Verfügbarkeit](#).

Das folgende Diagramm zeigt, wie es auf hoher Ebene Anwendungstests für AWS funktioniert.



Der Prozess besteht aus den folgenden Schritten:

1. Erstellen und verwalten Sie Testfälle, die kleinste Einheit von Testaktionen. Identifizieren Sie die Datentypen, die die funktionale Äquivalenz zwischen Quell- und Zielsystem am besten repräsentieren.
2. Definieren Sie die Konfiguration der Testumgebung, indem Sie CloudFormation Vorlagen und zusätzliche Attribute angeben.
3. Erstellen Sie Testsuiten, d. h. Sammlungen von Testfällen.
4. Datensätze hochladen und wiedergeben: Erfassen Sie die Eingabe- und Ausgabedatensätze auf dem Mainframe, laden Sie sie auf das Zielsystem hoch und spielen Sie das Testszenario anschließend erneut auf dem Zielsystem ab. AWS
5. Vergleichen Sie Quell- und Zieldatensätze. Anwendungstests für AWS vergleicht automatisch die Ausgabedatensätze des Quell- und Zielsystems. Überprüfen und bewerten Sie diese, um Unstimmigkeiten zu identifizieren.

Weitere Informationen finden Sie in der Dokumentation zu [AWS Mainframe Modernization](#).

Cutover

Eine der wichtigsten Herausforderungen bei der Mainframe-Modernisierung ist die Minimierung von Ausfallzeiten und Risiken beim Übergang zu einer neuen Plattform. Die blue/green Implementierungsstrategie bietet einen leistungsstarken und flexiblen Ansatz für die Systemmigration.

Bei der Blau/Grün-Implementierung handelt es sich um eine Technik, die Ausfallzeiten und Risiken reduziert, indem zwei identische Produktionsumgebungen ausgeführt werden, die als blaue und grüne Umgebungen bezeichnet werden. So funktioniert es im Zusammenhang mit der Mainframe-Modernisierung:

- **Blaue Umgebung:** Dies ist Ihr aktuelles Mainframe-System, das den gesamten Produktionsverkehr abwickelt.
- **Umweltfreundliche Umgebung:** Dies ist Ihre neue, modernisierte Plattform, die bereit ist AWS, die Kontrolle zu übernehmen.

Die blue/green Umstellungsstrategie umfasst folgende Schritte: Bereitstellung, Inbetriebnahme, Rollback, falls Probleme auftreten, und Abschluss.

Bereitstellung

In dieser Phase stellen Sie die neue (grüne) Umgebung bereit, AWS indem Sie die folgenden Schritte ausführen:

1. **Neue Plattform für die Umgebung:** Die [Route 53-Hosting-Zone](#) muss einen [DNS-Eintrag](#) enthalten, der auf die Mainframe-Umgebung verweist (blau).
2. **Überprüfen Sie die Konnektivität:** Stellen Sie sicher, dass die Verbindung zwischen Ihren AWS-Konto und den lokalen Transaktionsmanagern und Db2 for Database ordnungsgemäß ist. z/OS
3. **Führen Sie Rauchttests durch:** Verwenden Sie die AWS Load Balancer-Adresse, um auf die neue Plattformumgebung zuzugreifen, und führen Sie umfassende Rauchttests durch, um Folgendes zu überprüfen:
 - Alle erwarteten Workloads sind verfügbar.
 - 3270 Transaktionen werden korrekt verarbeitet.
 - Dateninteraktionen mit Db2 for z/OS funktionieren wie erwartet.

Gehen Sie live

In dieser Phase verlagern Sie den Verkehr in die grüne Umgebung und überwachen die Veränderungen.

1. Verwenden Sie die Verkehrsleitlinien in Route 53, um den Verkehr zu verlagern:
 - Option A: Sie können den gesamten Verkehr auf einmal verlagern.
 - Option B: Alternativ können Sie eine schrittweise gewichtete Verteilung verwenden.
2. Überwachen und validieren Sie:
 - Beobachten Sie die AWS Umgebung genau, wenn sich der Verkehr verschiebt.
 - Überprüfen Sie die 3270-Transaktionsverarbeitung.
 - Überprüfen Sie die Kommunikation mit Db2. z/OS
 - Achten Sie auf Leistungsprobleme.
 - Lassen Sie die Benutzer die Transaktionsergebnisse validieren.

Rollback

Falls Probleme auftreten, können Sie Route 53 schnell aktualisieren, um den Verkehr zurück in die lokale Mainframe-Umgebung (blau) umzuleiten.

Sie sollten die Probleme untersuchen und lösen, bevor Sie eine weitere Umstellung versuchen.

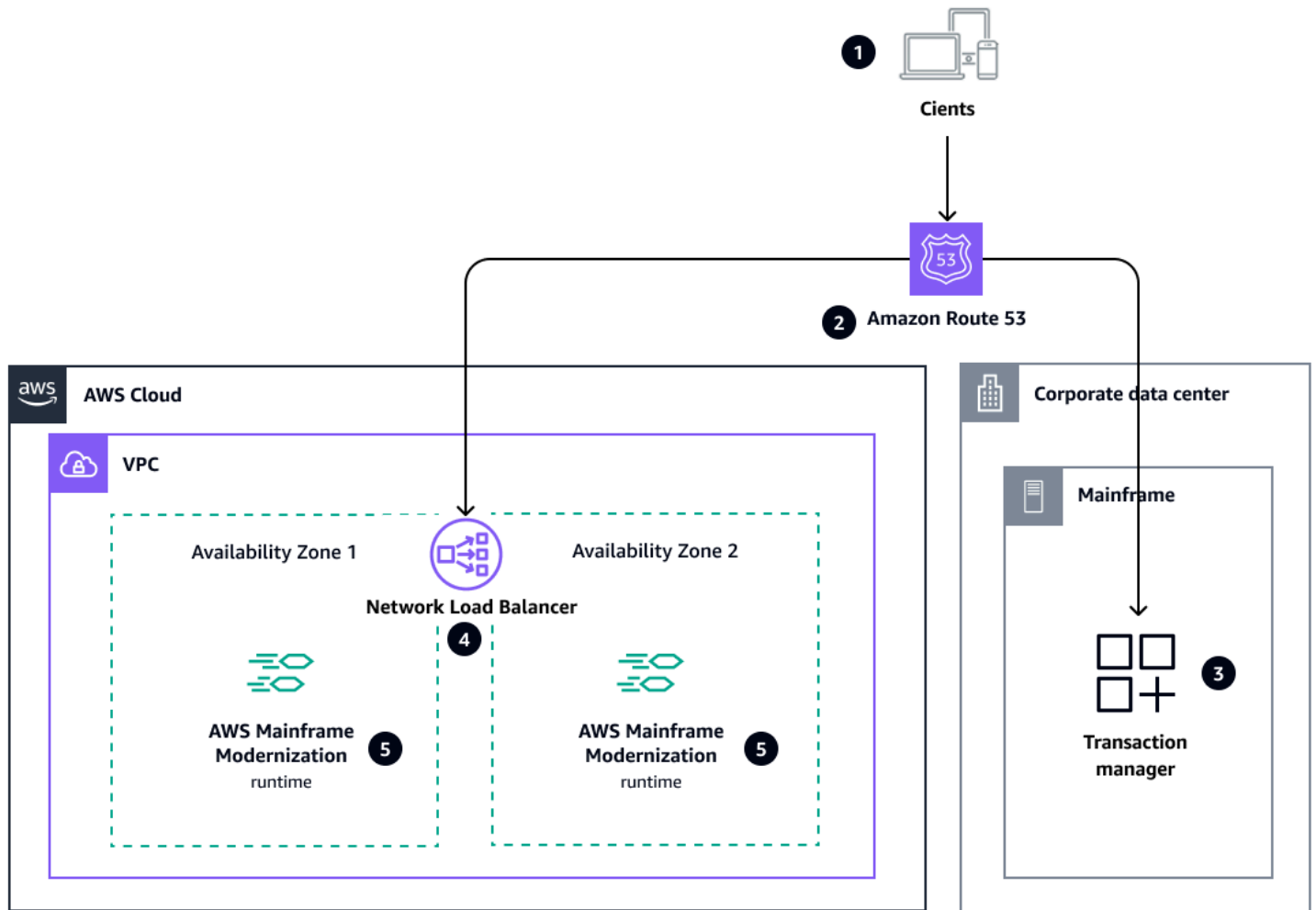
Schließen Sie ab

Nachdem Sie den Datenverkehr überwacht und überprüft haben, ob Ihre grüne Umgebung ordnungsgemäß funktioniert, können Sie den Anwendungsdatenverkehr schrittweise auf erhöhen AWS.

Nach einer stabilen Phase können Sie die Mainframe-Transaktionsumgebung (blau) außer Betrieb nehmen und die Datenbank Db2 for z/OS database vor Ort behalten.

Architektur

Das folgende Diagramm veranschaulicht den Ablauf der Umstellung.



Der Umstellungsprozess umfasst Folgendes:

1. Die Client-Anwendungen, Frontends und Backends für Frontends (BFFs) senden Transaktionen an den Route 53-Domainnamen.
2. Route 53 leitet die Verbindung je nach definierter Routing-Richtlinie zum Mainframe-Transaktionsmanager oder zum Network Load Balancer weiter.
3. Der Transaktionsmanager verarbeitet die Transaktionen, die an den Mainframe gesendet werden.
4. Der Network Load Balancer verteilt Transaktionen zur Verarbeitung an die verfügbaren Replattform-Umgebungen.
5. Die AWS Mainframe Modernization Umplattformumgebungen verarbeiten die Anfragen.

Bewährte Methoden

In diesem Abschnitt werden eine Reihe von bewährten Methoden zur Bewältigung der wichtigsten Herausforderungen bei der Umstellung von Mainframe-Workloads auf Cloud-Umgebungen unter Beibehaltung der Datenbank auf Db2 for z/OS beschrieben.

Netzwerklatenz

Um die Latenzauswirkungen einer Trennung der Anwendung von der Db2-Datenbank während einer Plattforummstellung genau vorhersagen zu können, empfehlen wir Ihnen, die Anzahl der Db2-Aufrufe sowohl für Transaktionen als auch für Batch-Prozesse gründlich zu bewerten. Diese Bewertung sollte anhand von Trace-Daten durchgeführt werden und die folgenden Schritte umfassen:

- Erfassung von Trace-Daten: Sammeln Sie detaillierte Traces von repräsentativen Transaktionen und Batch-Jobs und stellen Sie sicher, dass die Traces alle Db2-Interaktionen, einschließlich Ein- und Ausgängen, erfassen.
- Analysieren Sie die Trace-Daten: Zählen Sie die Anzahl der Db2-Ein- und Ausgänge für jede Transaktion und jeden Batch-Job und berechnen Sie die durchschnittliche Anzahl von Db2-Interaktionen pro Transaktion und Batch-Prozess.
- Messen Sie die aktuellen Antwortzeiten: Prüfen Sie, ob der Db2-Zugriff mit dem Service Level Agreement (SLA) Ihrer Anwendung übereinstimmt.
- Schätzen Sie die Netzwerklatenz ab: Ermitteln Sie die erwartete Netzwerklatenz zwischen der Anwendung auf die neue Plattform und der Db2-Datenbank. Berücksichtigen Sie Faktoren wie physische Entfernung, Netzwerkinfrastruktur und potenzielle Engpässe.
- Berechnen Sie die potenziellen Auswirkungen: Multiplizieren Sie für jede Transaktion und jeden Batch-Prozess die Anzahl der Db2-Ein- und -Ausgänge mit der geschätzten Netzwerklatenz. Fügen Sie diese berechnete Zeit zu den aktuellen Antwortzeiten hinzu, um die neue Gesamtverarbeitungszeit vorherzusagen.
- Beurteilen Sie die Ergebnisse: Beurteilen Sie, ob der prognostizierte Latenzanstieg für Ihre Geschäftsanforderungen akzeptabel ist, und identifizieren Sie alle Transaktionen oder Prozesse, die möglicherweise optimiert oder neu gestaltet werden müssen, um Latenzprobleme zu minimieren.
- Erwägen Sie Strategien zur Risikominderung: Erkunden Sie Optionen wie Verbindungspooling, Caching oder Batch-Datenabruf, um die Anzahl der einzelnen Db2-Interaktionen zu reduzieren.

Prüfen Sie die Möglichkeit, Daten, auf die häufig zugegriffen wird, näher an die Anwendungsebene zu verlagern.

Wenn Sie diese Schritte befolgen, können Sie datengestützte Entscheidungen über die Durchführbarkeit Ihrer Umstellungsstrategie treffen und potenzielle Leistungsprobleme erkennen, bevor sie sich auf Ihre Produktionsumgebung auswirken. Dieser Ansatz trägt dazu bei, einen reibungslosen Übergang zu gewährleisten und gleichzeitig ein akzeptables Leistungsniveau für Ihre datenbankabhängigen Anwendungen aufrechtzuerhalten.

Sicherheit

- Schützen Sie Ihren Anwendungsaufbau: Verwenden Sie für die Ausführung ein privates Subnetz in der Virtual Private Cloud (VPC), AWS CodeBuild um Isolation und erhöhte Sicherheit zu gewährleisten. Implementieren Sie einen vertrauenswürdigen Db2-Kontext aus dem CodeBuild Subnetz CIDR für den sicheren Datenbankzugriff während des Erstellungsprozesses.
- Schützen Sie Ihre Laufzeitumgebung: Verwenden Sie einen vertrauenswürdigen Db2-Kontext aus dem Laufzeitsubnetz CIDR für sichere Datenbankverbindungen.
- Sicheres Verwalten von Datenbankanmeldedaten: Implementieren Sie einen regelmäßigen Rotationsplan für die Anmeldeinformationen, um das Risiko eines unbefugten Zugriffs zu minimieren. Speichern Sie Db2-Anmeldeinformationen sicher in AWS Secrets Manager
- Sorgen Sie für Netzwerksicherheit: Implementieren Sie strenge Netzwerksegmentierungs- und Firewallregeln, um sowohl Build- als auch Laufzeitumgebungen zu schützen. Verwenden Sie die richtige Kombination von AWS Direct Connect und AWS Site-to-Site VPN , um das erforderliche Sicherheitsniveau zu erreichen.
- Verschlüsselung erzwingen: Erzwingen Sie die Verschlüsselung für Daten, die zwischen Ihrer Anwendung und Db2 for z/OS übertragen werden.

Anwendungs-Governance

- Etablieren Sie eine Informationsquelle: Richten Sie beispielsweise das neue Software Configuration Management (SCM) GitHub als zentrale Informationsquelle für den migrierten Anwendungscode ein. Dadurch wird Konsistenz gewährleistet und Versionsunterschiede zwischen der Cloud- und der Mainframe-Umgebung während der Übergangsphase vermieden.

- Aktualisieren Sie den Change-Management-Prozess: Aktualisieren Sie den Change-Management-Prozess, um Codeänderungen in diesem neuen Paradigma mit dualer Umgebung zu berücksichtigen. Dieser Prozess sollte Folgendes beinhalten:
 - Klare Genehmigungsworkflows für Codeänderungen.
 - Obligatorische Codeüberprüfungen vor der Zusammenführung des Codes in den Hauptzweig.
 - Synchronisierte Bereitstellungsverfahren, um sicherzustellen, dass beide Umgebungen gleichzeitig Updates erhalten.
 - Rollback-Mechanismen bei Problemen in einer der Umgebungen.

Elastizität

Die Elastizität von Cloud Computing führt zu einem Paradigmenwechsel, der die Kostenstruktur und das Ressourcenmanagement von Mainframes erheblich verändert. Im Gegensatz zur herkömmlichen Mainframe-Umgebung, die über feste Kapazitäten und Preismodelle nach Spitzenwerten verfügt, bieten Cloud-Plattformen dynamische Skalierbarkeit und einen pay-as-you-go Ansatz, der potenziell zu erheblichen Kosteneinsparungen und verbesserter betrieblicher Effizienz führen kann.

In einer Cloud-Umgebung können Unternehmen ihre Rechenressourcen je nach Bedarf in Echtzeit nach oben oder unten skalieren, wodurch die Notwendigkeit einer Überprovisionierung zur Bewältigung von Spitzenlasten entfällt. Diese Flexibilität ermöglicht es Unternehmen, nur für die Ressourcen zu zahlen, die sie verbrauchen, anstatt in teure Hardware- und Softwarelizenzen zu investieren, um gelegentliche Nutzungsspitzen zu bewältigen.

Einzelheiten zur Funktionsweise der Preisgestaltung finden Sie AWS unter [AWS Preisgestaltung](#).

Nächste Schritte

Die Mainframe-Modernisierung ist eine komplexe und wichtige Initiative, die Fachwissen und fortschrittliche Lösungen erfordert. Durch [strategische Partnerschaften](#), die Sie bei den folgenden Aufgaben unterstützen, können Sie Ihren Modernisierungsprozess beschleunigen und schnellere Geschäftsergebnisse erzielen:

- **Evaluieren und priorisieren:** Überprüfen Sie Ihre Mainframe-Anwendungen und ermitteln Sie, welche für eine Plattformumstellung geeignet sind, während die Datenbank weiterhin auf Db2 for z/OS läuft. Berücksichtigen Sie Faktoren wie Komplexität, geschäftliche Bedeutung und potenzielle Investitionsrendite (ROI).
- **Entwickeln Sie eine Migrationsstrategie:** Erstellen Sie einen detaillierten Plan für die Plattformumstellung Ihrer ausgewählten Anwendungen, einschließlich Zeitplänen, Ressourcenzuweisung und Strategien zur Risikominderung.
- **Bewerten Sie Tools und Technologien:** Suchen Sie nach geeigneten Tools und Technologien, um den Umstellungsprozess zu erleichtern, und wählen Sie sie aus, z. B. Plattformen zur Anwendungsmodernisierung oder Tools zur Codekonvertierung.
- **Arbeiten Sie mit Experten zusammen:** Ziehen Sie eine Partnerschaft mit Spezialisten für Mainframe-Modernisierung oder Beratungsunternehmen in Betracht, die Erfahrung mit Replattform-Projekten haben.
- **Machbarkeitsnachweis:** Beginnen Sie mit einem kleinen Machbarkeitsnachweis, um Ihren Ansatz zu validieren und potenzielle Herausforderungen zu identifizieren, bevor Sie auf größere Anwendungen skalieren.
- **Testen und Validieren:** Entwickeln Sie eine umfassende Teststrategie, um sicherzustellen, dass Ihre auf die neue Plattform umgestellten Anwendungen ordnungsgemäß funktionieren und die Datenintegrität mit Ihrer vorhandenen Db2-Datenbank gewahrt bleibt. z/OS
- **Schulung und Wissenstransfer:** Bereiten Sie Ihr Team auf die neue Umgebung vor, indem Sie Schulungen zu den Anwendungen auf der neuen Plattform und zu allen neu eingeführten Tools oder Technologien anbieten.
- **Schrittweise Implementierung:** Ziehen Sie einen schrittweisen Ansatz für die Umstellung der Plattform in Betracht, bei dem Sie Anwendungen schrittweise migrieren und gleichzeitig die Leistung überwachen und alle auftretenden Probleme beheben.

-
- **Kontinuierliche Optimierung:** Überwachen und optimieren Sie nach der Plattformierung kontinuierlich die Leistung Ihrer Anwendungen und deren Interaktionen mit der Db2 for Database, um den langfristigen Erfolg sicherzustellen. z/OS
 - **Modernisieren Sie in Ihrem Tempo:** Jetzt, da der Workload weiterläuft AWS und die Vorteile der Cloud bereits nutzt, beginnen Sie mit der Planung der Neukonzeptionsphase Ihrer Modernisierung.

Ressourcen

Weitere Informationen zur Mainframe-Migration und -Modernisierung finden Sie in den folgenden Ressourcen.

AWS Dokumentation

- [Konfigurieren von Amazon Route 53 als DNS-Service](#)
- [Weiterleitung des Datenverkehrs an einen ELB-Load Balancer](#)
- [Gewichtetes Routing](#)
- [Neuausrichtung von Anwendungen mit Rocket Software](#)

Referenzen von Rocket Software

- [Micro Focus Schnittstelle für externe Anrufe \(ECI\)](#)
- [CICS-Webdienste](#)

IBM-Referenzen

- [Vertrauenswürdige Kontexte](#) (IBM Db2 für die z/OS Dokumentation)

Tools

- [Rocket Enterprise Server](#)

AWS Präskriptive Leitlinien und Leitfäden

- [Erstellen Sie COBOL Db2-Programme mit und AWS Mainframe ModernizationAWS CodeBuild](#)
- [DevOps für AWS Mainframe Modernization](#)
- [Mainframe-Modernisierung: Entkopplungsmuster für die Migration von Anwendungscode](#)
- [Sichern und optimieren Sie den Benutzerzugriff in einer Db2-Verbunddatenbank mithilfe vertrauenswürdiger Kontexte AWS](#)

Dokumentverlauf

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

Änderung	Beschreibung	Datum
Erste Veröffentlichung	—	7. Mai 2025

AWS Glossar zu präskriptiven Leitlinien

Die folgenden Begriffe werden häufig in Strategien, Leitfäden und Mustern von AWS Prescriptive Guidance verwendet. Um Einträge vorzuschlagen, verwenden Sie bitte den Link Feedback geben am Ende des Glossars.

Zahlen

7 Rs

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

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

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

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

A

ABAC

Siehe [attributbasierte](#) Zugriffskontrolle.

abstrahierte Dienste

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.

Aggregatfunktion

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

AI

Siehe [künstliche Intelligenz](#).

AIOps

Siehe [Operationen im Bereich künstliche Intelligenz](#).

Anonymisierung

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

Anti-Muster

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

Anwendungssteuerung

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

Anwendungsportfolio

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

künstliche Intelligenz (KI)

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

Operationen mit künstlicher Intelligenz (AIOps)

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

Asymmetrische Verschlüsselung

Ein 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 die erfolgreiche Umstellung auf die Cloud unterstützt. AWS CAF unterteilt die Leitlinien in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Unternehmen, Mitarbeiter, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Aus dieser Perspektive bietet AWS CAF Leitlinien für Personalentwicklung, Schulung und Kommunikation, um das Unternehmen auf eine erfolgreiche

Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der [AWS -CAF-Webseite](#) und dem [AWS -CAF-Whitepaper](#).

AWS Workload-Qualifizierungsrahmen (AWS WQF)

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

B

schlechter Bot

Ein [Bot](#), der Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen soll.

BCP

Siehe [Planung der Geschäftskontinuität](#).

Verhaltensdiagramm

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

Big-Endian-System

Ein System, welches das höchstwertige Byte zuerst speichert. Siehe auch [Endianness](#).

Binäre Klassifikation

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

Bloom-Filter

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

Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

Bot

Eine Softwareanwendung, die automatisierte Aufgaben über das Internet ausführt und menschliche Aktivitäten oder Interaktionen simuliert. Manche Bots sind nützlich oder nützlich, wie z. B. Webcrawler, die Informationen im Internet indexieren. Einige andere Bots, 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 normalerweise keine Zugriffsrechte besitzt. Weitere Informationen finden Sie unter dem Indikator [Implementation break-glass procedures](#) in den AWS Well-Architected-Leitlinien.

Brownfield-Strategie

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

Puffer-Cache

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

Geschäftsfähigkeit

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

Planung der Geschäftskontinuität (BCP)

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

C

CAF

[Weitere Informationen finden Sie unter Framework AWS für die Cloud-Einführung.](#)

Bereitstellung auf Kanaren

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

CCoE

Weitere Informationen finden Sie [im Cloud Center of Excellence](#).

CDC

Siehe [Erfassung von Änderungsdaten](#).

Erfassung von Datenänderungen (CDC)

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

Chaos-Technik

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

CI/CD

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

Klassifizierung

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

clientseitige Verschlüsselung

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

Cloud-Exzellenzzentrum (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 — Tätigen Sie grundlegende Investitionen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer landing zone, Definition eines CCo E, Einrichtung eines Betriebsmodells)
- Migration – Migrieren einzelner Anwendungen
- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

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

CMDB

Siehe [Datenbank für das Konfigurationsmanagement](#).

Code-Repository

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

Kalter Cache

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

Kalte Daten

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

Computer Vision (CV)

Ein Bereich der [KI](#), der maschinelles Lernen nutzt, um Informationen aus visuellen Formaten wie digitalen Bildern und Videos zu analysieren und zu extrahieren. Amazon SageMaker AI bietet beispielsweise Bildverarbeitungsalgorithmen für CV.

Drift in der Konfiguration

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

Verwaltung der Datenbankkonfiguration (CMDB)

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

Konformitätspaket

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

Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

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

CV

Siehe [Computer Vision](#).

D

Daten im Ruhezustand

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

Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter [Datenklassifizierung](#).

Datendrift

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

Daten während der Übertragung

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

Datennetz

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

Datenminimierung

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

Datenperimeter

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

Vorverarbeitung der Daten

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

Herkunft der Daten

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

betreffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

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

defense-in-depth

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

delegierter Administrator

In AWS Organizations kann ein kompatibler Dienst ein AWS Mitgliedskonto registrieren, um die Konten der Organisation und die Berechtigungen für diesen Dienst zu verwalten. Dieses Konto wird als delegierter Administrator für diesen Service bezeichnet. Weitere Informationen und eine Liste kompatibler Services finden Sie unter [Services, die mit AWS Organizations funktionieren](#) in der AWS Organizations -Dokumentation.

Einsatz

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

Entwicklungsumgebung

Siehe [Umgebung](#).

Detektivische Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, ein Ereignis zu erkennen, zu protokollieren und zu warnen, nachdem ein Ereignis eingetreten ist. Diese Kontrollen stellen eine zweite Verteidigungslinie dar und warnen Sie vor 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, wie z. B. unbeabsichtigte Fehlkonfigurationen oder ein Malware-Angriff.

Notfallwiederherstellung (DR)

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

DML

Siehe Sprache zur [Datenbankmanipulation](#).

Domainorientiertes Design

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

DR

Siehe [Disaster Recovery](#).

Erkennung von Driften

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

DVSM

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

E

EDA

Siehe [explorative Datenanalyse](#).

EDI

Siehe [elektronischer Datenaustausch](#).

Edge-Computing

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

elektronischer Datenaustausch (EDI)

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

Verschlüsselung

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

Verschlüsselungsschlüssel

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

Endianismus

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

Endpunkt

[Siehe](#) Service-Endpunkt.

Endpunkt-Services

Ein Service, den Sie in einer Virtual Private Cloud (VPC) hosten können, um ihn mit anderen Benutzern zu teilen. Sie können einen 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. Bei Aufgaben, die spezifische Formatierungs-, Argumentations- oder Fachkenntnisse erfordern, kann die Eingabeaufforderung mit wenigen Handgriffen effektiv sein. [Siehe auch Zero-Shot Prompting](#).

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](#).

G

Generative KI

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

Geoblocking

Siehe [geografische Einschränkungen](#).

Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden,

um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in [der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte](#). CloudFront

Gitflow-Workflow

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

goldenes Bild

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

Greenfield-Strategie

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

Integritätsschutz

Eine allgemeine Regel, die dazu beiträgt, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Unternehmenseinheiten zu regeln (OUs). Präventiver Integritätsschutz setzt Richtlinien durch, um die Einhaltung von Standards zu gewährleisten. Sie werden mithilfe von Service-Kontrollrichtlinien und IAM-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.

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.

Daten zurückhalten

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

Homogene Datenbankmigration

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

heiße Daten

Daten, auf die häufig zugegriffen wird, z. B. Echtzeitdaten oder aktuelle 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 [Infrastruktur als Code](#).

Identitätsbasierte Richtlinie

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

Leerlaufanwendung

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

IIoT

Siehe [Industrielles Internet der Dinge](#).

unveränderliche Infrastruktur

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

Eingehende (ingress) VPC

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

Inkrementelle Migration

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

Industrie 4.0

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

Infrastruktur

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

Infrastructure as Code (IaC)

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

industrielles Internet der Dinge (T) Ilo

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

Inspektions-VPC

In einer Architektur AWS mit mehreren Konten eine zentralisierte VPC, die Inspektionen des Netzwerkverkehrs zwischen VPCs (in demselben oder unterschiedlichen AWS-Regionen), dem Internet und lokalen Netzwerken verwaltet. In der [AWS Security Reference Architecture](#) wird empfohlen, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr sowie Inspektionen einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Internet of Things (IoT)

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

Interpretierbarkeit

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

IoT

Siehe [Internet der Dinge](#).

IT information library (ITIL, IT-Informationsbibliothek)

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

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-Service management](#).

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.

SCHWARZ

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](#).

Mechanismus

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

Mitgliedskonto

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

MES

Siehe [Manufacturing Execution System](#).

Message Queuing-Telemetrietransport (MQTT)

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

Microservice

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

Microservices-Architekturen

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

Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf

die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

Migration in großem Maßstab

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

Migrationsfabrik

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

Migrationsmetadaten

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

Migrationsmuster

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

Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung,

Migrationspriorisierung und Wellenplanung). Das [MPA-Tool](#) (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

Migration Readiness Assessment (MRA)

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

Migrationsstrategie

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

ML

[Siehe maschinelles Lernen.](#)

Modernisierung

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

Bewertung der Modernisierungsfähigkeit

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

Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder

Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter [Zerlegen von Monolithen in Microservices](#).

MPA

Siehe [Bewertung des Migrationsportfolios](#).

MQTT

Siehe [Message Queuing-Telemetrietransport](#).

Mehrklassen-Klassifizierung

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

veränderbare Infrastruktur

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

O

OAC

Siehe [Origin Access Control](#).

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

Vereinbarung auf Betriebsebene (OLA)

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

Überprüfung der Betriebsbereitschaft (ORR)

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

Betriebstechnologie (OT)

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

Betriebsintegration (OI)

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

Organisationspfad

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

Organisatorisches Veränderungsmanagement (OCM)

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

Ursprungszugriffskontrolle (OAC)

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

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

P

Berechtigungsgrenze

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

persönlich identifizierbare Informationen (PII)

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

Personenbezogene Daten

Siehe [persönlich identifizierbare Informationen](#).

Playbook

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

PLC

Siehe [programmierbare Logiksteuerung](#).

PLM

Siehe [Produktlebenszyklusmanagement](#).

policy

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

Polyglotte Beharrlichkeit

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

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. Diese Entität ist in der Regel ein Root-Benutzer für eine AWS-Konto, eine IAM-Rolle oder einen Benutzer. Weitere Informationen finden Sie unter Prinzipal in [Rollenbegriffe und -konzepte](#) in der IAM-Dokumentation.

Datenschutz von Natur aus

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

Privat gehostete Zonen

Ein Container, der Informationen darüber enthält, wie Amazon Route 53 auf DNS-Abfragen für eine Domain und deren Subdomains innerhalb einer oder mehrerer VPCs Domains antworten soll. Weitere Informationen finden Sie unter [Arbeiten mit privat gehosteten Zonen](#) in der Route-53-Dokumentation.

proaktive Steuerung

Eine [Sicherheitskontrolle](#), die den Einsatz nicht richtlinienkonformer Ressourcen verhindern soll. Diese Steuerelemente scannen Ressourcen, bevor sie bereitgestellt werden. Wenn die Ressource nicht der Kontrolle entspricht, wird sie nicht bereitgestellt. Weitere Informationen finden Sie im [Referenzhandbuch zu Kontrollen](#) in der AWS Control Tower Dokumentation und unter [Proaktive Kontrollen](#) unter Implementierung von Sicherheitskontrollen am AWS.

Produktlebenszyklusmanagement (PLM)

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

Produktionsumgebung

Siehe [Umgebung](#).

Speicherprogrammierbare Steuerung (SPS)

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

schnelle Verkettung

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

Pseudonymisierung

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

publish/subscribe (pub/sub)

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

Q

Abfrageplan

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

Abfrageplanregression

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

R

RACI-Matrix

Siehe [verantwortlich, rechenschaftspflichtig, konsultiert, informiert \(RACI\)](#).

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

Reaktive Kontrolle

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

Beibehaltung

Siehe [7 Rs](#).

zurückziehen

Siehe [7 Rs](#).

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 Benutzeranmeldeinformationen, die Sie in verschlüsselter Form speichern. Es besteht aus dem geheimen Wert und seinen Metadaten. Der geheime Wert kann binär, eine einzelne Zeichenfolge oder mehrere Zeichenketten sein. Weitere Informationen finden Sie unter [Was ist in einem Secrets Manager Manager-Geheimnis?](#) in der Secrets Manager Manager-Dokumentation.

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

Service-Endpoint

Die URL des Einstiegspunkts für einen AWS-Service. Sie können den Endpoint 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](#).

SIEM

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

Single Point of Failure (SPOF)

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

SLA

Siehe [Service Level Agreement](#).

SLI

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

ALSO

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

split-and-seed Modell

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

SPOTTEN

Siehe [Single Point of Failure](#).

Sternschema

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

Strangler-Fig-Muster

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

Subnetz

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

Aufsichtskontrolle und Datenerfassung (SCADA)

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

Symmetrische Verschlüsselung

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

synthetisches Testen

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

Systemaufforderung

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

T

tags

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

Zielvariable

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

Aufgabenliste

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

Testumgebungen

[Siehe Umgebung.](#)

Training

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

Transit-Gateway

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

Stammbasierter Workflow

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

Vertrauenswürdiger Zugriff

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

Optimieren

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

Zwei-Pizzen-Team

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

U

Unsicherheit

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

undifferenzierte Aufgaben

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

höhere Umgebungen

Siehe [Umgebung](#).

V

Vacuuming

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

Versionskontrolle

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

VPC-Peering

Eine Verbindung zwischen zwei VPCs, die es Ihnen ermöglicht, den Verkehr mithilfe privater IP-Adressen weiterzuleiten. Weitere Informationen finden Sie unter [Was ist VPC-Peering?](#) in der Amazon-VPC-Dokumentation.

Schwachstelle

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

W

Warmer Cache

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

warme Daten

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

Fensterfunktion

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

Workload

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

Workstream

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

WURM

Sehen [Sie einmal schreiben, viele lesen](#).

WQF

Siehe [AWS Workload-Qualifizierungsrahmen](#).

einmal schreiben, viele lesen (WORM)

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

Z

Zero-Day-Exploit

Ein Angriff, in der Regel Malware, der eine [Zero-Day-Sicherheitslücke](#) ausnutzt.

Zero-Day-Sicherheitslücke

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

Eingabeaufforderung ohne Zwischenfälle

Bereitstellung von Anweisungen für die Ausführung einer Aufgabe an einen [LLM](#), jedoch ohne Beispiele (Schnappschüsse), die ihm als Orientierungshilfe dienen könnten. Der LLM muss sein vortrainiertes Wissen einsetzen, um die Aufgabe zu bewältigen. Die Effektivität von Zero-Shot Prompting hängt von der Komplexität der Aufgabe und der Qualität der Aufforderung ab. [Siehe auch Few-Shot-Prompting](#).

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.