Aktivierung der Datenpersistenz in Microservices

AWS Präskriptive Leitlinien

AWS Präskriptive Leitlinien: Aktivierung der Datenpersistenz in Microservices

Table of Contents

Aktivieren der Datenpersistenz in Microservices

Tabby Ward und Balaji Mohan, Amazon Web Services (AWS)

Dezember 2023 (Dokumentverlauf)

Organisationen suchen ständig nach neuen Prozessen, um Wachstumsmöglichkeiten zu schaffen und die Markteinführungszeit zu verkürzen. Sie können die Agilität und Effizienz Ihrer Organisation erhöhen, indem Sie Ihre Anwendungen, Software und IT-Systeme modernisieren. Modernisierung hilft Ihnen auch dabei, Ihren Kunden schnellere und bessere Services zu bieten.

Die Modernisierung von Anwendungen ist ein Gateway für die kontinuierliche Verbesserung Ihrer Organisation und beginnt mit der Umgestaltung einer monolithischen Anwendung in eine Reihe unabhängig entwickelter, bereitgestellter und verwalteter Microservices. Dieser Prozess umfasst die folgenden Schritte:

- Zerlegen von Monolithen in Microservices – Verwenden Sie Muster, um monolithische Anwendungen in Microservices aufzuteilen.
- Microservices integrieren – Integrieren Sie die neu erstellten Microservices in eine Microservices-Architektur mithilfe von AWSServerless Services von Amazon Web Services ().
- Aktivieren der Datenpersistenz für die Microservices-Architektur – Fördern Sie die Polyglot-Persistenz zwischen Ihren Microservices, indem Sie deren Datenspeicher dezentralisieren.

Obwohl Sie für einige Anwendungsfälle eine monolithische Anwendungsarchitektur verwenden können, funktionieren moderne Anwendungsfunktionen oft nicht in einer monolithischen Architektur. Beispielsweise kann die gesamte Anwendung nicht verfügbar bleiben, während Sie einzelne Komponenten aktualisieren, und Sie können einzelne Komponenten nicht skalieren, um Engpässe oder Schwachstellen zu beheben (relativ dichte Regionen in den Daten Ihrer Anwendung). Monolithe können zu großen, nicht überschaubaren Anwendungen werden, und zwischen mehreren Teams sind erhebliche Bemühungen und Koordination erforderlich, um kleine Änderungen einzuführen.

Legacy-Anwendungen verwenden in der Regel eine zentrale monolithische Datenbank, die Schemaänderungen schwierig macht, eine Technologiesperre mit vertikaler Skalierung als einzige Möglichkeit, auf Wachstum zu reagieren, und einen einzigen Fehlerpunkt vorsieht. Eine monolithische Datenbank verhindert auch, dass Sie die dezentralisierten und unabhängigen Komponenten erstellen, die für die Implementierung einer Microservices-Architektur erforderlich sind.

Zuvor bestand ein typischer Architekturansatz darin, alle Benutzeranforderungen in einer relationalen Datenbank zu modellieren, die von der monolithischen Anwendung verwendet wurde. Dieser Ansatz wurde von der beliebten relationalen Datenbankarchitektur unterstützt, und Anwendungsarchitekten haben das relationale Schema normalerweise in den frühesten Phasen des Entwicklungsprozesses entwickelt, ein hoch normalisiertes Schema erstellt und es dann an das Entwicklerteam gesendet. Dies bedeutete jedoch, dass die Datenbank das Datenmodell für den Anwendungsanwendungsfall bewegte, anstatt auf die andere Weise.

Indem Sie sich dafür entscheiden, Ihre Datenspeicher zu dezentralisieren, fördern Sie die Polyglot-Persistenz zwischen Ihren Microservices und identifizieren Ihre Datenspeichertechnologie anhand der Datenzugriffsmuster und anderer Anforderungen Ihrer Microservices. Jeder Microservice hat seinen eigenen Datenspeicher und kann unabhängig mit Schemaänderungen mit geringen Auswirkungen skaliert werden. Die Daten werden über die API des Microservice verwaltet. Die Aufteilung einer monolithischen Datenbank ist nicht einfach, und eine der größten Herausforderungen besteht darin, Ihre Daten so zu strukturieren, dass sie die bestmögliche Leistung erzielen. Die Persistenz dekodierter Polyglots führt in der Regel auch zu letztendlicher Datenkonsistenz, und andere potenzielle Herausforderungen, die eine gründliche Auswertung erfordern, umfassen die Datensynchronisierung während Transaktionen, die Transaktionsintegrität, die Datenduplizierung sowie Joins und Latenz.

Dieser Leitfaden richtet sich an Anwendungseigentümer, Geschäftsinhaber, Architekten, technische Leiter und Projektmanager. Das Handbuch enthält die folgenden sechs Muster, um die Datenpersistenz zwischen Ihren Microservices zu ermöglichen:

- D-atabase-per-service Muster
- API-Kompositionsmuster
- CQRS-Muster
- Ereignis-Sourcing-Muster
- Saga Muster
  - Schritte zur Implementierung des saga Musters mithilfe von AWS Step Functionsfinden Sie im Muster Implementieren des Serverless-sagaMusters mithilfe von AWS Step Functions auf der Website AWS Prescriptive Guidance.
- S-hared-database-per-service Muster

Der Leitfaden ist Teil einer Inhaltsreihe, die den von empfohlenen Ansatz zur Anwendungsmoderation behandeltAWS. Die Serie umfasst auch:

- Strategie zur Modernisierung von Anwendungen in der AWS Cloud
- Schrittweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud
- Bewertung der Modernisierungsbereitschaft für Anwendungen in der AWS Cloud
- Zerlegen von Monolithen in Microservices
- Integrieren von Microservices mithilfe von AWS Serverless-Services

Gezielte Geschäftsergebnisse

Viele Organisationen stellen fest, dass die Innovation und Verbesserung der Benutzererfahrung durch monolithische Anwendungen, Datenbanken und Technologien negativ beeinflusst wird. Ältere Anwendungen und Datenbanken reduzieren Ihre Optionen für die Einführung moderner Technologie-Frameworks und beschränken Ihre Wettbewerbsfähigkeit und Innovation. Wenn Sie Anwendungen und ihre Datenspeicher modernisieren, sind sie jedoch einfacher zu skalieren und schneller zu entwickeln. Eine entkoppelte Datenstrategie verbessert die Fehlertoleranz und Resilienz, was dazu beiträgt, die Markteinführungszeit für Ihre neuen Anwendungsfunktionen zu beschleunigen.

Sie sollten die folgenden sechs Ergebnisse erwarten, die sich aus der Förderung der Datenpersistenz zwischen Ihren Microservices ergeben:

- Entfernen Sie ältere monolithische Datenbanken aus Ihrem Anwendungsportfolio.
- Verbessern Sie Fehlertoleranz, Ausfallsicherheit und Verfügbarkeit für Ihre Anwendungen.
- Verkürzen Sie Ihre Zeit bis zur Markteinführung für neue Anwendungsfunktionen.
- Reduzieren Sie Ihre gesamten Lizenzkosten und Betriebskosten.
- Nutzen Sie Open-Source-Lösungen (z. B. MySQL oder PostgreSQL).
- Erstellen Sie hochgradig skalierbare und verteilte Anwendungen, indem Sie aus mehr als 15 speziell entwickelten Datenbank-Engines in der AWS Cloud wählen.

Muster zum Aktivieren der Datenpersistenz

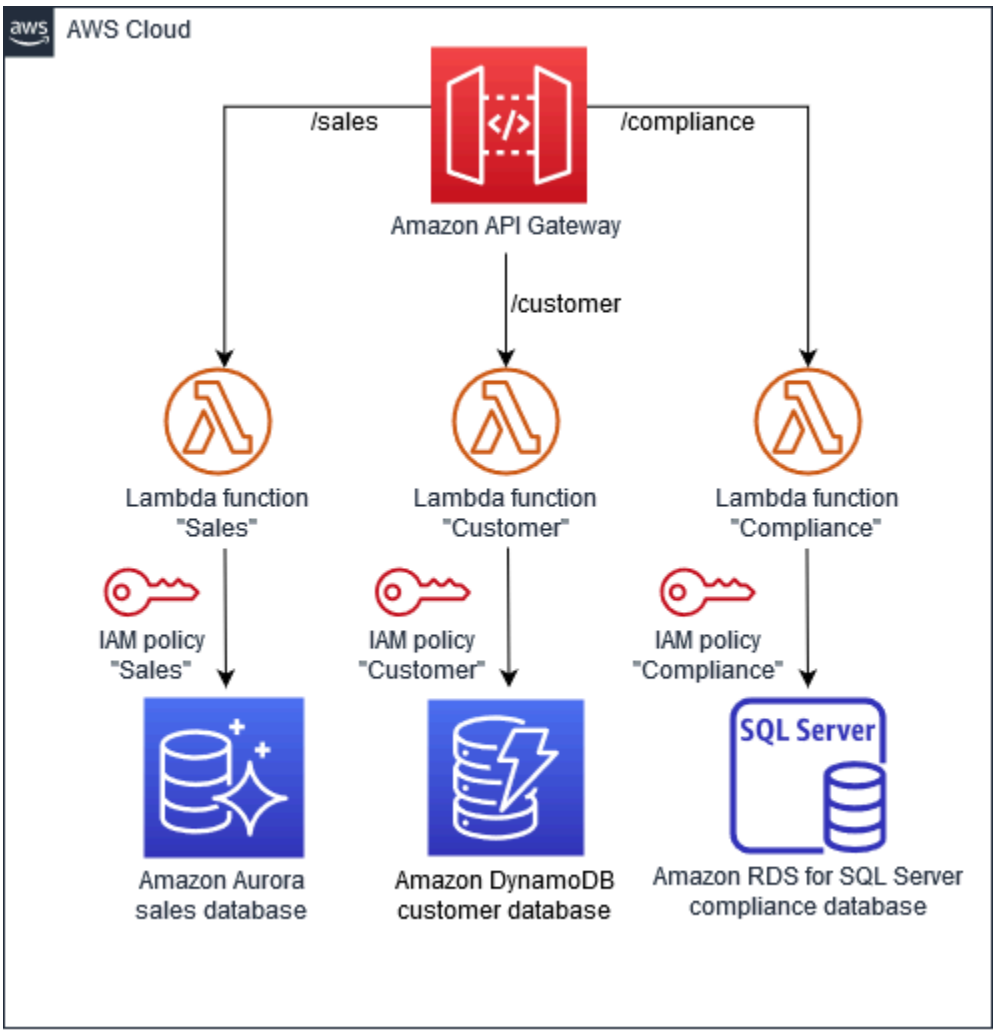
Die folgenden Muster werden verwendet, um die Datenpersistenz in Ihren Microservices zu aktivieren.

Themen

D-atabase-per-service Muster

Die Trichterverbindung ist das Kernmerkmal einer Microservices-Architektur, da jeder einzelne Microservice Informationen unabhängig voneinander speichern und aus seinem eigenen Datenspeicher abrufen kann. Durch die Bereitstellung des database-per-service Musters wählen Sie die am besten geeigneten Datenspeicher (z. B. relationale oder nicht relationale Datenbanken) für Ihre Anwendung und Geschäftsanforderungen aus. Das bedeutet, dass Microservices keine Datenschicht gemeinsam nutzen, Änderungen an der individuellen Datenbank eines Microservice keine Auswirkungen auf andere Microservices haben, dass auf einzelne Datenspeicher nicht direkt von anderen Microservices zugegriffen werden kann und dass auf persistente Daten nur von APIs zugegriffen wird. Das Entkoppeln von Datenspeichern verbessert auch die Ausfallsicherheit Ihrer gesamten Anwendung und stellt sicher, dass eine einzelne Datenbank nicht ein einziger Fehlerpunkt sein kann.

In der folgenden Abbildung werden verschiedene AWS Datenbanken von den Microservices „Vertrieb“, „Kunde“ und „Compliance“ verwendet. Diese Microservices werden als -AWS LambdaFunktionen bereitgestellt und über eine Amazon API Gateway-API aufgerufen. AWS Identity and Access Management (IAM)-Richtlinien stellen sicher, dass Daten privat gehalten und nicht zwischen den Microservices geteilt werden. Jeder Microservice verwendet einen Datenbanktyp, der seine individuellen Anforderungen erfüllt. Beispielsweise verwendet „Vertrieb“ Amazon Aurora, „Kunde“ verwendet Amazon DynamoDB und „Compliance“ verwendet Amazon Relational Database Service (Amazon RDS) für SQL Server.



Sie sollten erwägen, dieses Muster zu verwenden, wenn:

- Zwischen Ihren Microservices ist eine Trichterverbindung erforderlich.
- Microservices haben unterschiedliche Compliance- oder Sicherheitsanforderungen für ihre Datenbanken.
- Eine genauere Skalierungssteuerung ist erforderlich.

Die Verwendung des database-per-service Musters hat die folgenden Nachteile:

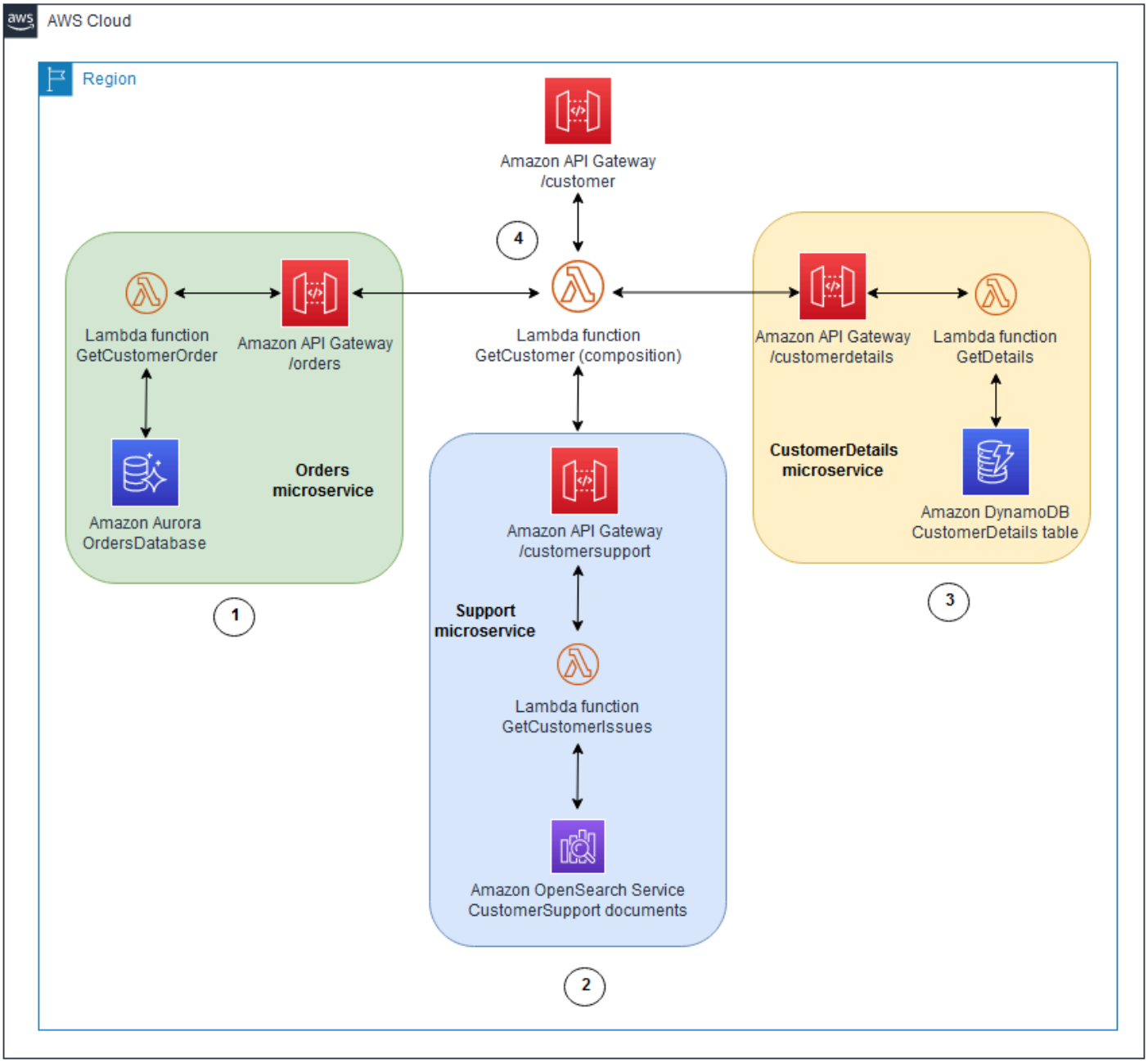
- Es kann schwierig sein, komplexe Transaktionen und Abfragen zu implementieren, die sich über mehrere Microservices oder Datenspeicher erstrecken.
- Sie müssen mehrere relationale und nicht relationale Datenbanken verwalten.
- Ihre Datenspeicher müssen zwei der Anforderungen des CAP-Theors erfüllen: Konsistenz , Verfügbarkeit oder Partitionstoleranz .

> **Note**
>
> Wenn Sie das database-per-service Muster verwenden, müssen Sie die API-Kompositionsmuster oder die bereitstellen, CQRS-Muster um Abfragen zu implementieren, die sich über mehrere Microservices erstrecken.

API-Kompositionsmuster

Dieses Muster verwendet einen API-Composer oder Aggregator, um eine Abfrage zu implementieren, indem einzelne Microservices aufgerufen werden, die Eigentümer der Daten sind. Anschließend werden die Ergebnisse kombiniert, indem ein In-Memory-Join durchgeführt wird.

Das folgende Diagramm veranschaulicht, wie dieses Muster implementiert wird.



Das Diagramm zeigt den folgenden Workflow:

1. Ein API-Gateway bedient die "/customer"-API, die über einen "Orders"-Microservice verfügt, der Kundenaufträge in einer Aurora-Datenbank verfolgt.
2. Der Microservice „Support“ verfolgt Probleme mit dem Kundensupport und speichert sie in einer Amazon- OpenSearch Service-Datenbank.

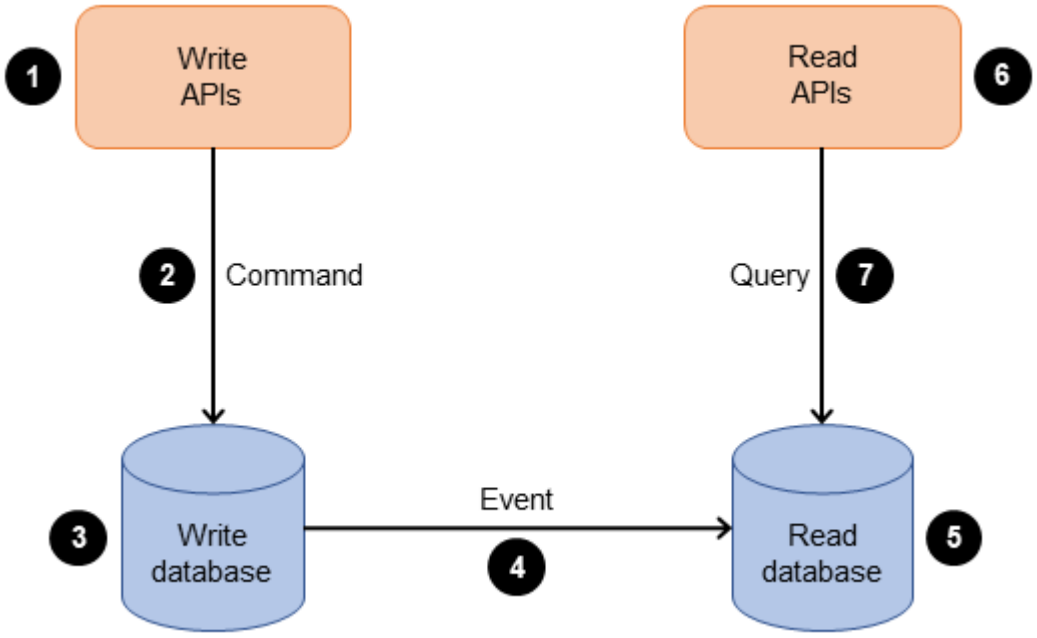
3. Der Microservice „CustomerDetails“ verwaltet Kundenattribute (z. B. Adresse, Telefonnummer oder Zahlungsdetails) in einer DynamoDB-Tabelle.
4. Die „GetCustomer“-Lambda-Funktion führt die APIs für diese Microservices aus und führt einen speicherinternen Join für die Daten durch, bevor sie an den Anforderer zurückgegeben werden. Dies hilft, Kundeninformationen in einem Netzwerkaufruf an die benutzerorientierte API einfach abzurufen, und hält die Schnittstelle sehr einfach.

Das API-Kompositionsmuster bietet die einfachste Möglichkeit, Daten aus mehreren Microservices zu sammeln. Die Verwendung des API-Kompositionsmusters hat jedoch die folgenden Nachteile:

- Es eignet sich möglicherweise nicht für komplexe Abfragen und große Datensätze, die In-Memory-Joins erfordern.
- Ihr Gesamtsystem ist weniger verfügbar, wenn Sie die Anzahl der Microservices erhöhen, die mit dem API-Composer verbunden sind.
- Erhöhte Datenbankanforderungen erzeugen mehr Netzwerkverkehr, was Ihre Betriebskosten erhöht.

CQRS-Muster

Das CQRS-Muster (Command Query Responsibility segregation) trennt die Datenmutation oder den Befehlsteil eines Systems vom Abfrageteil. Sie können das CQRS-Muster verwenden, um Updates und Abfragen zu trennen, wenn sie unterschiedliche Anforderungen an Durchsatz, Latenz oder Konsistenz haben. Das CQRS-Muster teilt die Anwendung in zwei Teile auf – die Befehlsseite und die Abfrageseite –, wie im folgenden Diagramm gezeigt. Die Befehlsseite verarbeitet `create`-`update`, -und -`delete`Anforderungen. Die Abfrageseite führt den `query` Teil mithilfe der Lesereplikate aus.



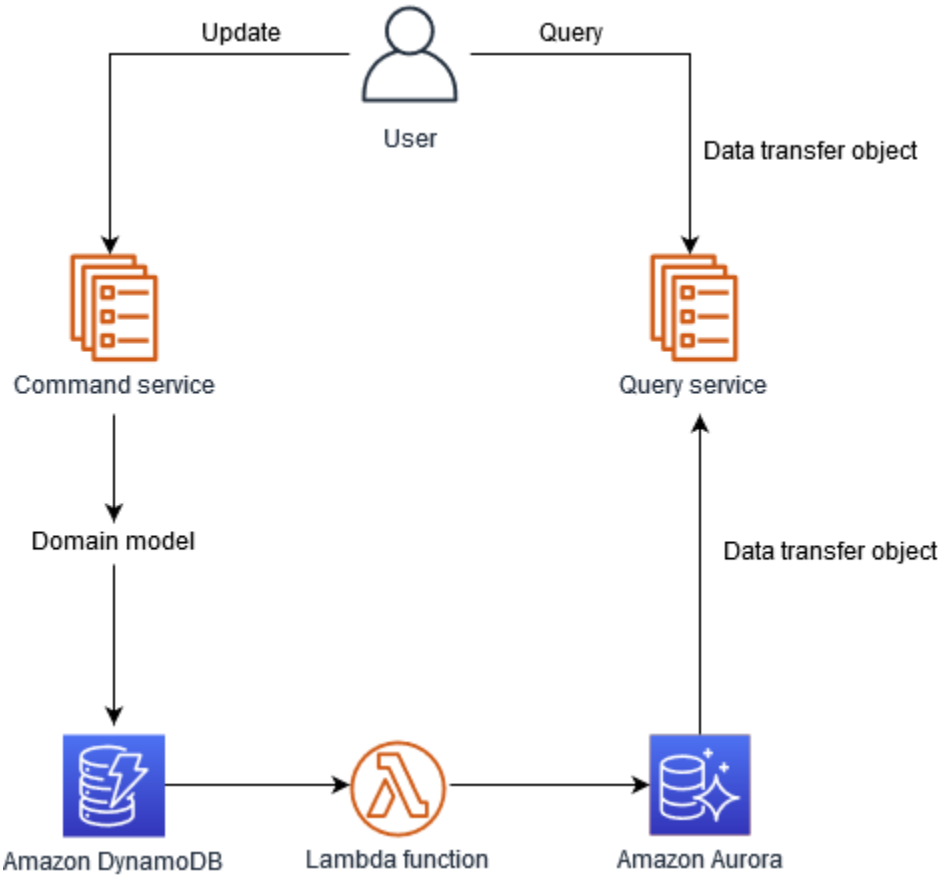
Das Diagramm zeigt den folgenden Prozess:

1. Das Unternehmen interagiert mit der Anwendung, indem es Befehle über eine API sendet. Befehle sind Aktionen wie das Erstellen, Aktualisieren oder Löschen von Daten.
2. Die Anwendung verarbeitet den eingehenden Befehl auf der Befehlsseite. Dazu gehört die Validierung, Autorisierung und Ausführung des Vorgangs.
3. Die Anwendung behält die Daten des Befehls in der Schreibdatenbank (Befehl) bei.
4. Nachdem der Befehl in der Schreibdatenbank gespeichert wurde, werden Ereignisse ausgelöst, um die Daten in der Lese- (Abfrage-)Datenbank zu aktualisieren.
5. Die Lese- (Abfrage-)Datenbank verarbeitet und behält die Daten bei. Lesedatenbanken sind für spezifische Abfrageanforderungen optimiert.
6. Das Unternehmen interagiert mit Lese-APIs, um Abfragen an die Abfrageseite der Anwendung zu senden.
7. Die Anwendung verarbeitet die eingehende Abfrage auf der Abfrageseite und ruft die Daten aus der Lesedatenbank ab.

Sie können das CQRS-Muster implementieren, indem Sie verschiedene Kombinationen von Datenbanken verwenden, darunter:

- Verwenden von relationalen Datenbankmanagementsystem (RDBMS)-Datenbanken sowohl für den Befehl als auch für die Abfrageseite. Schreibvorgänge gehen an die Primärdatenbank und Lesevorgänge können an Lesereplikate weitergeleitet werden. Beispiel: Amazon-RDS-Lesereplikate
- Verwenden einer RDBMS-Datenbank für die Befehlsseite und einer NoSQL-Datenbank für die Abfrageseite. Beispiel: Modernisieren älterer Datenbanken mithilfe von Event Sourcing und CQRS mit AWS DMS
- Verwenden von NoSQL-Datenbanken sowohl für den Befehl als auch für die Abfrageseite. Beispiel: Erstellen eines CQRS-Ereignisspeichers mit Amazon DynamoDB
- Verwenden einer NoSQL-Datenbank für die Befehlsseite und einer RDBMS-Datenbank für die Abfrageseite, wie im folgenden Beispiel beschrieben.

In der folgenden Abbildung wird ein NoSQL-Datenspeicher wie DynamoDB verwendet, um den Schreibdurchsatz zu optimieren und flexible Abfragefunktionen bereitzustellen. Dadurch wird eine hohe Schreibskalierbarkeit für Workloads erreicht, die beim Hinzufügen von Daten klar definierte Zugriffsmuster aufweisen. Eine relationale Datenbank wie Amazon Aurora bietet komplexe Abfragefunktionen. Ein DynamoDB-Stream sendet Daten an eine Lambda-Funktion, die die Aurora-Tabelle aktualisiert.



Die Implementierung des CQRS-Musters mit DynamoDB und Aurora bietet folgende wichtige Vorteile:

- DynamoDB ist eine vollständig verwaltete NoSQL-Datenbank, die Schreibvorgänge mit hohem Volumen verarbeiten kann, und Aurora bietet hohe Leseskalierbarkeit für komplexe Abfragen auf der Abfrageseite.
- DynamoDB bietet Zugriff mit niedriger Latenz und hohem Durchsatz auf Daten, was es ideal für die Verarbeitung von Befehls- und Aktualisierungsvorgängen macht, und die Aurora-Leistung kann für komplexe Abfragen optimiert und optimiert werden.
- Sowohl DynamoDB als auch Aurora bieten Serverless-Optionen, mit denen Ihr Unternehmen nur für Ressourcen zahlen kann, die auf der Nutzung basieren.
- DynamoDB und Aurora sind vollständig verwaltete Services, wodurch der betriebliche Aufwand für die Verwaltung von Datenbanken, Backups und Skalierbarkeit reduziert wird.

Sie sollten die Verwendung des CQRS-Musters in Betracht ziehen, wenn:

- Sie haben das database-per-service Muster implementiert und möchten Daten aus mehreren Microservices verknüpfen.
- Ihre Lese- und Schreib-Workloads haben separate Anforderungen an Skalierung, Latenz und Konsistenz.
- Die letztendliche Konsistenz ist für die Leseabfragen akzeptabel.

> **Important**
>
> Das CQRS-Muster führt in der Regel zu einer letztendlichen Konsistenz zwischen den Datenspeichern.

Ereignis-Sourcing-Muster

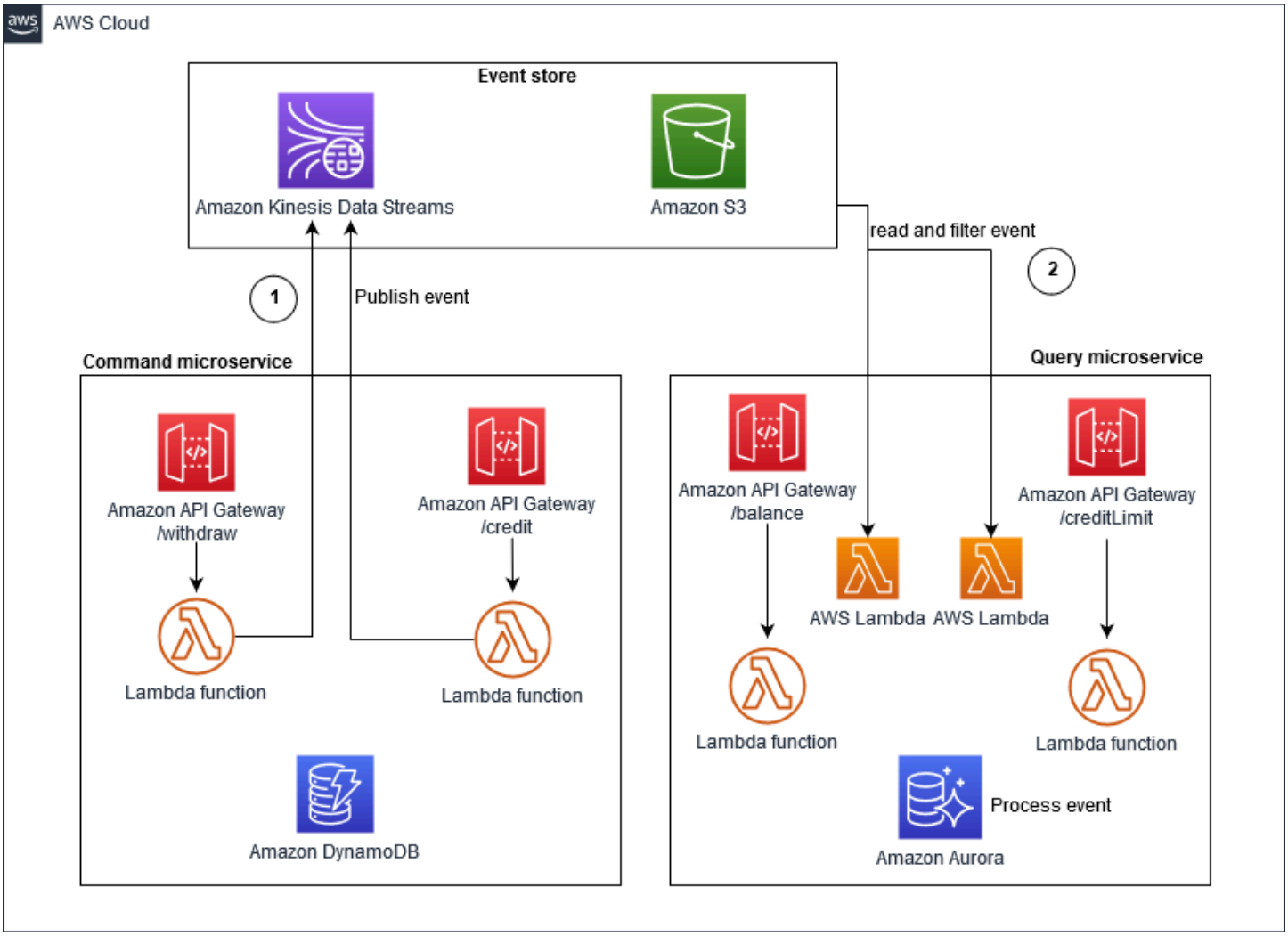
Das Ereignis-Sourcing-Muster wird in der Regel mit verwendet, CQRS-Muster um Lese- und Schreib-Workloads zu entkoppeln und Leistung, Skalierbarkeit und Sicherheit zu optimieren. Daten werden als eine Reihe von Ereignissen gespeichert, anstatt Datenspeicher direkt zu aktualisieren. Microservices wiederholen Ereignisse aus einem Ereignisspeicher, um den entsprechenden Status ihrer eigenen Datenspeicher zu berechnen. Das Muster bietet Einblick in den aktuellen Zustand der Anwendung und zusätzlichen Kontext dafür, wie die Anwendung in diesen Zustand gekommen ist. Das Ereignis-Sourcing-Muster funktioniert effektiv mit dem CQRS-Muster, da Daten für ein bestimmtes Ereignis reproduziert werden können, auch wenn die Befehls- und Abfragedatenspeicher unterschiedliche Schemata haben.

Durch Auswahl dieses Musters können Sie den Status der Anwendung jederzeit identifizieren und rekonstruieren. Dies erzeugt einen persistenten Audit-Trail und erleichtert das Debuggen. Daten werden jedoch letztendlich konsistent und dies ist für einige Anwendungsfälle möglicherweise nicht angemessen.

Dieses Muster kann entweder mithilfe von Amazon Kinesis Data Streams oder Amazon implementiert werden EventBridge.

Implementierung von Amazon Kinesis Data Streams

In der folgenden Abbildung ist Kinesis Data Streams die Hauptkomponente eines zentralen Ereignisspeichers. Der Ereignisspeicher erfasst Anwendungsänderungen als Ereignisse und behält sie auf Amazon Simple Storage Service (Amazon S3) bei.



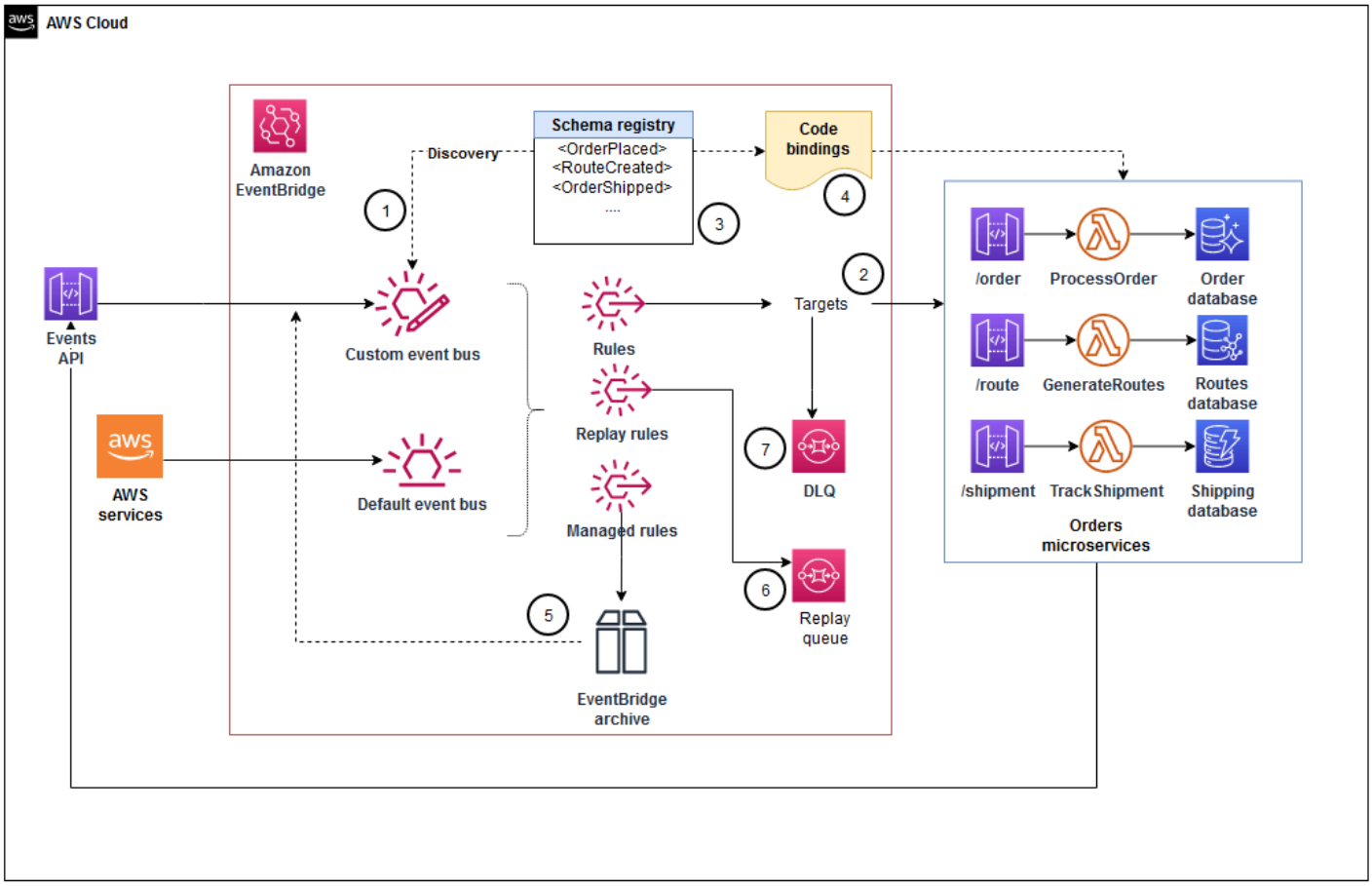
Der Workflow besteht aus folgenden Schritten:

1. Wenn bei den Microservices „/withdraw“ oder „/credit“ eine Änderung des Ereignisstatus auftritt, veröffentlichen sie ein Ereignis, indem sie eine Nachricht in Kinesis Data Streams schreiben.
2. Andere Microservices wie „/balance“ oder „/creditLimit“ lesen eine Kopie der Nachricht, filtern sie nach Relevanz und leiten sie zur weiteren Verarbeitung weiter.

Amazon- EventBridge Implementierung

Die Architektur in der folgenden Abbildung verwendet EventBridge. EventBridge ist ein Serverless-Service, der Ereignisse verwendet, um Anwendungskomponenten zu verbinden, was es Ihnen erleichtert, skalierbare, ereignisgesteuerte Anwendungen zu erstellen. Bei der ereignisgesteuerten Architektur werden lose gekoppelte Softwaresysteme erstellt, die zusammenarbeiten, indem sie Ereignisse ausgeben und darauf reagieren. EventBridge bietet einen Standard-Event-

Bus für Ereignisse, die von -AWSServices veröffentlicht werden, und Sie können auch einen benutzerdefinierten Event Bus für domainspezifische Buses erstellen.



Der Workflow besteht aus folgenden Schritten:

1. „OrderPlaced“-Ereignisse werden vom Microservice „Bestellungen“ im benutzerdefinierten Event Bus veröffentlicht.
2. Microservices, die nach der Bestellung Maßnahmen ergreifen müssen, z. B. der Microservice „/route“, werden durch Regeln und Ziele initiiert.
3. Diese Microservices generieren eine Route, um die Bestellung an den Kunden zu senden und ein „RouteCreated“-Ereignis auszugeben.
4. Microservices, die weitere Maßnahmen ergreifen müssen, werden auch durch das Ereignis „RouteCreated“ initiiert.
5. Ereignisse werden an ein Ereignisarchiv gesendet (z. B. EventBridge archivieren), sodass sie bei Bedarf zur erneuten Verarbeitung wiedergegeben werden können.

6. Ereignisse der historischen Reihenfolge werden bei Bedarf zur erneuten Verarbeitung an eine neue Amazon SQS-Warteschlange (Wiederholungswarteschlange) gesendet.
7. Wenn Ziele nicht initiiert werden, werden die betroffenen Ereignisse zur weiteren Analyse und erneuten Verarbeitung in eine Warteschlange für unzustellbare Nachrichten (DLQ) gestellt.

Sie sollten erwägen, dieses Muster zu verwenden, wenn:

- Ereignisse werden verwendet, um den Status der Anwendung vollständig neu zu erstellen.
- Sie müssen Ereignisse im System wiederholen und der Status einer Anwendung kann jederzeit bestimmt werden.
- Sie möchten in der Lage sein, bestimmte Ereignisse rückgängig zu machen, ohne mit einem leeren Anwendungsstatus beginnen zu müssen.
- Ihr System benötigt einen Stream von Ereignissen, die einfach serialisiert werden können, um ein automatisiertes Protokoll zu erstellen.
- Ihr System erfordert hohe Lesevorgänge, ist jedoch leicht bei Schreibvorgängen; hohe Lesevorgänge können an eine speicherinterne Datenbank weitergeleitet werden, die mit dem Ereignis-Stream aktualisiert wird.

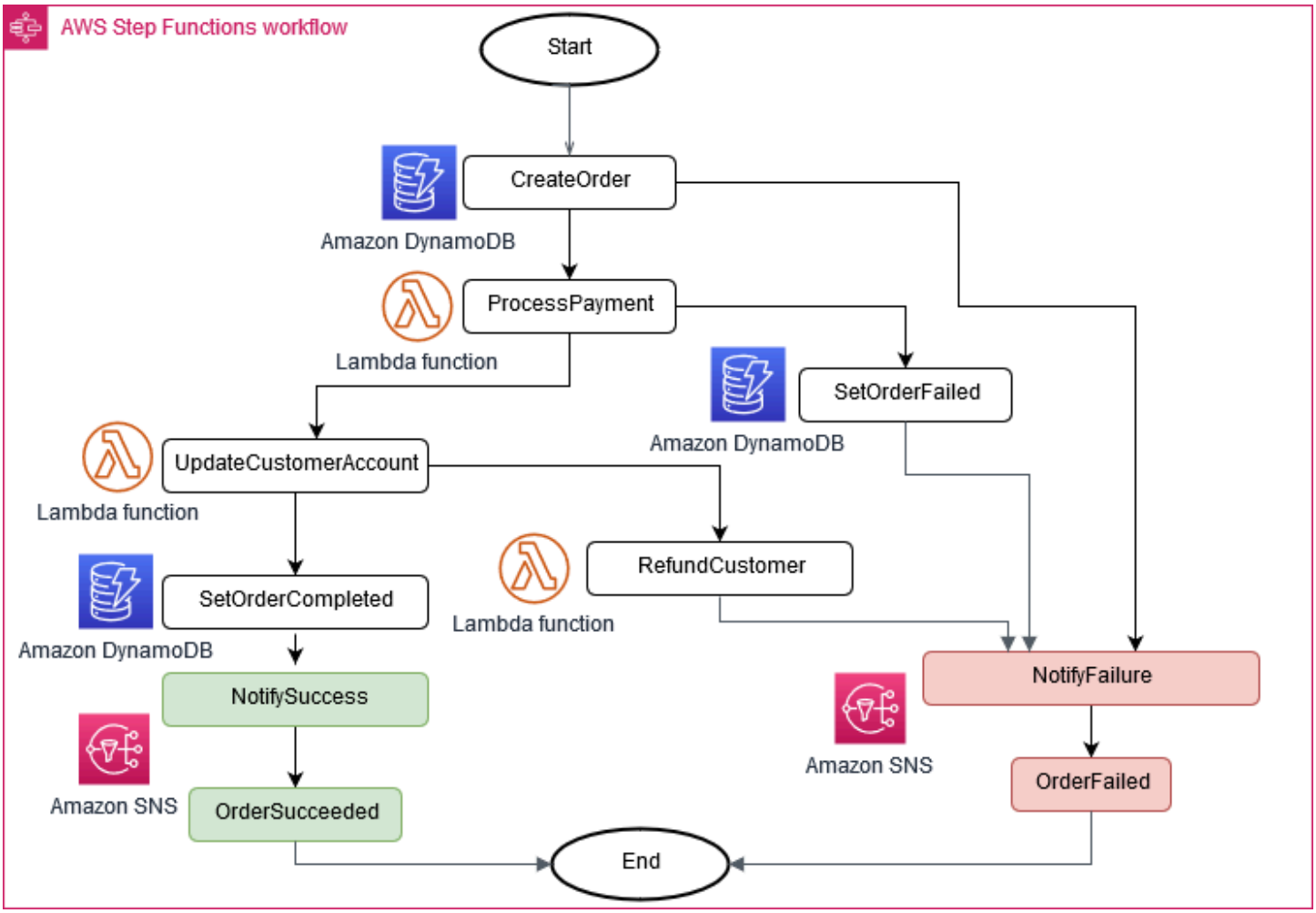
> **Important**
>
> Wenn Sie das Ereignis-Sourcing-Muster verwenden, müssen Sie bereitstellen, Saga Muster um die Datenkonsistenz über Microservices hinweg aufrechtzuerhalten.

Saga Muster

Das saga Muster ist ein Fehlerverwaltungsmuster, das dazu beiträgt, die Konsistenz verteilter Anwendungen herzustellen und Transaktionen zwischen mehreren Microservices zu koordinieren, um die Datenkonsistenz aufrechtzuerhalten. Ein Microservice veröffentlicht ein Ereignis für jede Transaktion, und die nächste Transaktion wird basierend auf dem Ergebnis des Ereignisses initiiert. Je nach Erfolg oder Misserfolg der Transaktionen können zwei verschiedene Pfade verwendet werden.

Die folgende Abbildung zeigt, wie das saga Muster ein Bestellverarbeitungssystem mithilfe von implementiertAWS Step Functions. Jeder Schritt (z. B. „ProcessPayment“) hat auch separate

Schritte, um den Erfolg (z. B. „UpdateCustomerAccount“) oder Misserfolg (z. B. „SetOrderFailure“) des Prozesses zu bewältigen.



Sie sollten erwägen, dieses Muster zu verwenden, wenn:

- Die Anwendung muss die Datenkonsistenz über mehrere Microservices hinweg aufrechterhalten, ohne dass es zu einer strikten Beeinträchtigung kommt.
- Es gibt langlebige Transaktionen und Sie möchten nicht, dass andere Microservices blockiert werden, wenn ein Microservice lange läuft.
- Sie müssen ein Rollback durchführen können, wenn ein Vorgang in der Sequenz fehlschlägt.

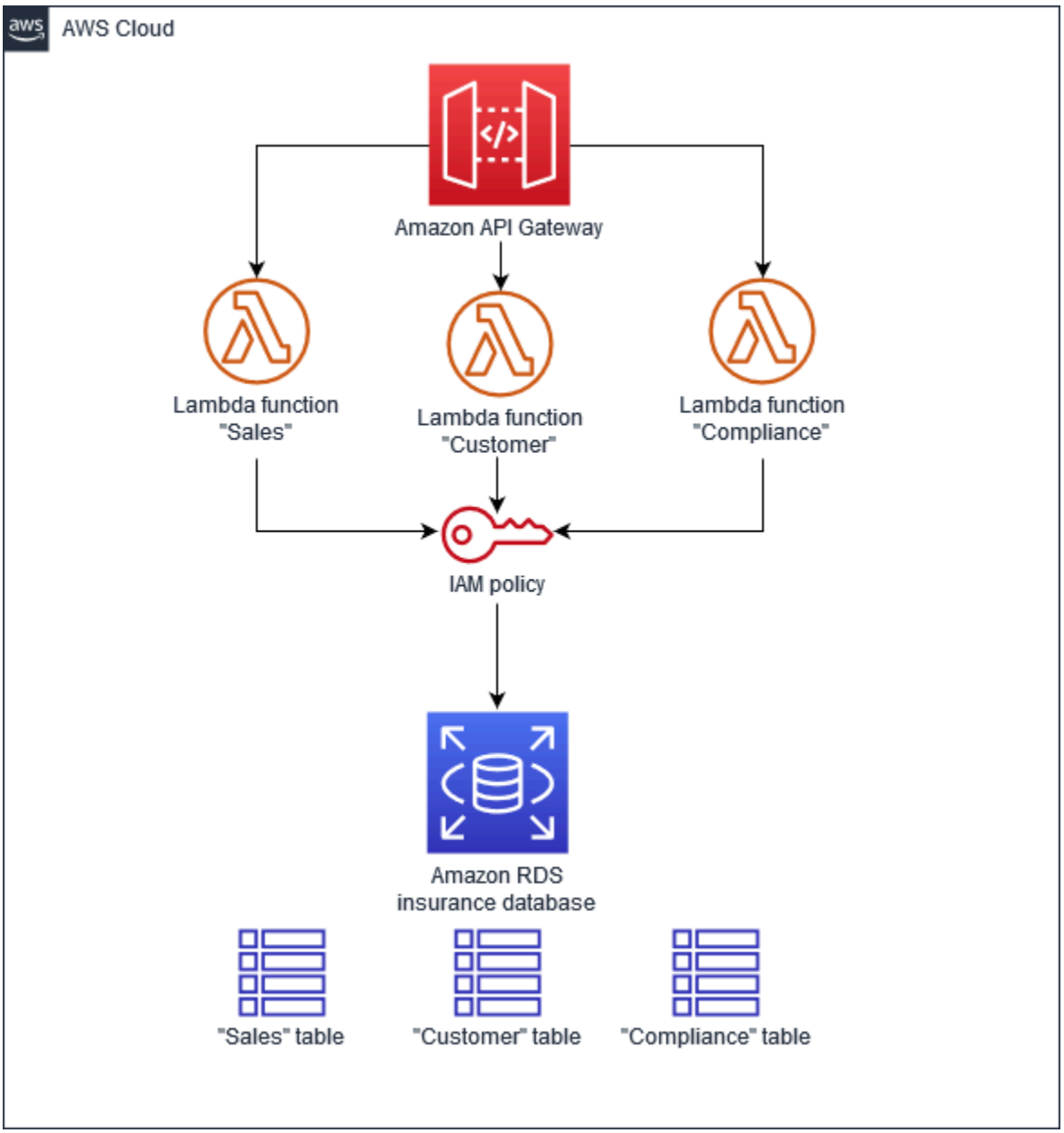
> **Important**
>
> Das saga Muster ist schwer zu debuggen und seine Komplexität nimmt mit der Anzahl der Microservices zu. Das Muster erfordert ein komplexes Programmiermodell, das Transaktionen für Rollback und Rückgängigmachen von Änderungen entwickelt und entwirft.

Weitere Informationen zur Implementierung des saga Musters in einer Microservices-Architektur finden Sie im Muster Implementieren Sie das Serverless-sagaMuster mithilfe von AWS Step Functions auf der Website AWS Prescriptive Guidance.

S-hared-database-per-service Muster

Im shared-database-per-service Muster wird dieselbe Datenbank von mehreren Microservices gemeinsam genutzt. Sie müssen die Anwendungsarchitektur sorgfältig bewerten, bevor Sie dieses Muster anwenden, und sicherstellen, dass Sie Hot-Tabellen (einzelne Tabellen, die zwischen mehreren Microservices gemeinsam genutzt werden) vermeiden. Alle Ihre Datenbankänderungen müssen auch abwärtskompatibel sein. Entwickler können beispielsweise Spalten oder Tabellen nur löschen, wenn nicht von der aktuellen und der vorherigen Version aller Microservices auf Objekte verwiesen wird.

In der folgenden Abbildung wird eine Versicherungsdatenbank von allen Microservices gemeinsam genutzt und eine IAM-Richtlinie bietet Zugriff auf die Datenbank. Dies sorgt für eine arbeitsfreie Entwicklungszeit, z. B. muss eine Änderung des Microservice „Vertrieb“ Schemaänderungen mit dem Microservice „Kunde“ koordinieren. Dieses Muster reduziert nicht die Abhängigkeiten zwischen den Entwicklungsteams und führt die Laufzeiteinbindung ein, da alle Microservices dieselbe Datenbank nutzen. Beispielsweise können lang andauernde „Sales“-Transaktionen die „Customer“-Tabelle sperren und so die „Customer“-Transaktionen blockieren.



Sie sollten erwägen, dieses Muster zu verwenden, wenn:

- Sie möchten nicht zu viel Faktorwechsel Ihrer vorhandenen Codebasis vornehmen.
- Sie erzwingen die Datenkonsistenz, indem Sie Transaktionen verwenden, die Atomizität, Konsistenz, Isolation und Haltbarkeit (ACID) bereitstellen.
- Sie möchten nur eine Datenbank warten und betreiben.
- Die Implementierung des database-per-service Musters ist aufgrund von Abhängigkeiten zwischen Ihren vorhandenen Microservices schwierig.
- Sie möchten Ihre vorhandene Datenebene nicht vollständig neu entwerfen.

Häufig gestellte Fragen

Dieser Abschnitt enthält Antworten auf häufig aufgeworfene Fragen zur Aktivierung der Datenpersistenz in Microservices.

Wann kann ich meine monolithische Datenbank im Rahmen meiner Modernisierungsreise modernisieren?

Sie sollten sich auf die Modernisierung Ihrer monolithischen Datenbank konzentrieren, wenn Sie beginnen, monolithische Anwendungen in Microservices zu zerlegen. Stellen Sie sicher, dass Sie eine Strategie erstellen, um Ihre Datenbank in mehrere kleine Datenbanken aufzuteilen, die an Ihren Anwendungen ausgerichtet sind.

Kann ich eine monolithische Legacy-Datenbank für mehrere Microservices behalten?

Die Beibehaltung einer freigegebenen monolithischen Datenbank für mehrere Microservices führt zu einer engen Kopplung, was bedeutet, dass Sie Änderungen an Ihren Microservices nicht unabhängig voneinander bereitstellen können und dass alle Schemaänderungen zwischen Ihren Microservices koordiniert werden müssen. Obwohl Sie einen relationalen Datenspeicher als monolithische Datenbank verwenden können, sind NoSQL-Datenbanken möglicherweise eine bessere Wahl für einige Ihrer Microservices.

Was sollte ich beim Entwerfen von Datenbanken für eine Microservices-Architektur beachten?

Sie sollten Ihre Anwendung basierend auf Domänen entwerfen, die mit den Funktionen Ihrer Anwendung übereinstimmen. Stellen Sie sicher, dass Sie die Funktionalität der Anwendung auswerten und entscheiden, ob ein relationales Datenbankschema erforderlich ist. Sie sollten auch erwägen, eine NoSQL-Datenbank zu verwenden, wenn sie Ihren Anforderungen entspricht.

Was ist ein gängiges Muster zur Aufrechterhaltung der Datenkonsistenz über verschiedene Microservices hinweg?

Das gebräuchlichste Muster ist die Verwendung einesereignisgesteuerte Architekturaus.

Wie halte ich die Transaktionsautomatisierung aufrecht?

In einer Microservices-Architektur besteht eine Transaktion aus mehreren lokalen Transaktionen, die von verschiedenen Microservices abgewickelt werden. Wenn eine lokale Transaktion fehlschlägt, müssen Sie die zuvor abgeschlossenen erfolgreichen Transaktionen rückgängig machen. Sie können dasSaga Musterum dies zu vermeiden.

Muss ich für jeden Microservice eine separate Datenbank verwenden?

Der Hauptvorteil einer Microservices-Architektur ist die lose Kopplung. Die persistenten Daten jedes Microservices müssen privat und nur über die API eines Microservices zugänglich gehalten werden. Änderungen am Datenschema müssen sorgfältig ausgewertet werden, wenn Ihre Microservices dieselbe Datenbank teilen.

Wie kann ich die persistenten Daten eines Microservices privat halten, wenn sie alle eine einzige Datenbank teilen?

Wenn Ihre Microservices eine relationale Datenbank teilen, stellen Sie sicher, dass Sie private Tabellen für jeden Microservice haben. Sie können auch einzelne Schemas erstellen, die für die einzelnen Microservices privat sind.

Ressourcen

Verwandte Anleitungen und Muster

- Strategie zur Modernisierung von Anwendungen in derAWSWolke
- Schrittweiser Ansatz zur Modernisierung von Anwendungen in derAWSWolke
- Bewertung der Modernisierungsbereitschaft für Anwendungen in derAWSWolke
- Zerlegung von Monolithen in Microservices
- Integration von Microservices mithilfe von AWS-Services ohne Server
- Implementieren der ServerlesssagaPattern unter Verwendung vonAWS Step Functions

Sonstige Ressourcen

- Anwendungsmodernisierung mitAWS
- Erstellen Sie hochverfügbare Microservices für Anwendungen jeder Größe und Größenordnung
- Cloud-native Anwendungsmodernisierung mitAWS
- Kostenoptimierung und Innovation: Eine Einführung in die Anwendungsmodernisierung
- -Entwicklerhandbuch: Skalieren mit Microservices
- Verteilte Datenverwaltung —SagaPattern
- Implementierung von Microservice-Architekturen mithilfe von AWS-Services: Muster für die Trennung der Verantwortlichkeit von Befeh
- Implementierung von Microservice-Architekturen mithilfe von AWS-Services: Ereignismuster
- Moderne Anwendungen: Wertschöpfung durch Anwendungsdesign
- Modernisieren Sie Ihre Anwendungen, fördern Sie das Wachstum und senken Sie die TCO

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
Aktualisiertes Muster	Wir haben den Abschnitt Amazon- EventBridge Implementierung des Ereignis-Sourcing-Musters aktualisiert.	4. Dezember 2023
Erweiterter Abschnitt	Wir haben das CQRS-Muster mit weiteren Informationen aktualisiert.	17. November 2023
Link zur Implementierung des saga Musters mit Step Functions hinzugefügt	Wir haben die Abschnitte Home und Saga pattern mit dem Link zum pattern Implement the serverless saga pattern by using AWS Step Functions von der Website AWS Prescriptive Guidance aktualisiert.	23. Februar 2021
Erste Veröffentlichung	—	27. Januar 2021

Glossar zu AWS Prescriptive Guidance

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

Zahlen

7 Rs

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

- Faktorwechsel/Architekturwechsel – Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile cloudnativer Feature nutzen, um Agilität, Leistung und Skalierbarkeit zu verbessern. Dies beinhaltet in der Regel die Portierung des Betriebssystems und der Datenbank. Beispiel: Migrieren Sie Ihre On-Premises-Oracle-Datenbank zu der PostgreSQL-kompatible Amazon-Aurora-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 On-Premises-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 On-Premises-Oracle-Datenbank zu Oracle auf einer EC2-Instance 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. Dieses Migrationsszenario ist spezifisch für VMware Cloud in AWS, welches die Kompatibilität mit virtuellen Maschinen (VM) und die Workload-Portabilität zwischen Ihrer On-Premises-Umgebung und AWS unterstützt. Sie können die VMware-Cloud-Foundation-Technologien von Ihren On-Premises-Rechenzentren aus verwenden, wenn

Sie Ihre Infrastruktur zu VMware Cloud in AWS migrieren. Beispiel: Verschieben Sie den Hypervisor, der Ihre Oracle-Datenbank hostet, auf VMware Cloud in 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 .

Abstrakte Services

Siehe -verwaltete Services.

ACID

Siehe Atomizität, Konsistenz, Isolation, 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. Sie ist flexibler, erfordert aber mehr Arbeit als die Aktiv-Passiv-Migration.

Aktiv-Passiv-Migration

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

Aggregatfunktion

Eine SQL-Funktion, die für eine 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 für künstliche Intelligenz.

Anonymisierung

Der Prozess des dauerhaften Löschens personenbezogener Daten in einem Datensatz. Anonymisierung kann dazu beitragen, den persönlichen Datenschutz zu schützen. Anonymisierte Daten werden nicht mehr als personenbezogene Daten betrachtet.

Anti-Muster

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

-Anwendungskontrolle

Ein Sicherheitsansatz, der nur die Verwendung genehmigter Anwendungen ermöglicht, um ein System vor Malware zu schützen.

Anwendungsportfolio

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

künstliche Intelligenz (KI)

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

Operationen mit künstlicher Intelligenz (AIOps)

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

Asymmetrische Verschlüsselung

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

Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

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

Attributbasierte Zugriffskontrolle (ABAC)

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

Autorisierende Datenquelle

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

Availability Zone

Ein eigener Standort in einer AWS-Region, der isoliert und somit vor Ausfällen in anderen Availability Zones geschützt ist und kostengünstige Netzwerkkonnektivität mit geringer Latenz zu anderen Availability Zones in der gleichen Region bereitstellt.

AWS Cloud Adoption Framework (AWS CAF)

Ein Rahmen von Leitlinien und bewährten Verfahren von AWS, um Organisationen bei der Entwicklung eines effizienten und effektiven Plans für eine erfolgreiche Umstellung auf die Cloud zu unterstützen. AWS CAF unterteilt die Beratung in sechs Schwerpunktbereiche, die

als Perspektiven bezeichnet werden: Geschäft, Menschen, 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. Für diese Perspektive bietet AWS-CAF Beratung für Personalentwicklung, Training und Kommunikation, um die Organisation auf eine erfolgreiche Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der AWS-CAF-Webseite und dem AWS-CAF-Whitepaper.

AWS Workload Qualification Framework (AWS WQF)

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

B

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 Endianität .

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.

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 Verzweigungen (GitHub Dokumentation).

Break-Glass-Zugriff

Unter außergewöhnlichen Umständen und durch einen genehmigten Prozess ist dies eine schnelle Möglichkeit für einen Benutzer, Zugriff auf einen zu erhaltenAWS-Konto, für den er normalerweise keine Zugriffsberechtigungen besitzt. Weitere Informationen finden Sie im Indikator Implement break-glass procedures in der AWS Well-Architected-Anleitung.

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

Siehe AWS Cloud Adoption Framework.

CCoE

Weitere Informationen finden Sie unter Cloud-Kompetenzzentrum.

CDC

Siehe Erfassung von Datenänderungen.

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-Engineering

Absichtliche Einführung von Ausfällen oder störenden Ereignissen, um die Ausfallsicherheit eines Systems zu testen. Sie können AWS Fault Injection Service (AWS FIS) verwenden, um Experimente durchzuführen, die Ihre AWSWorkloads belasten, und deren Antwort bewerten.

CI/CD

Siehe kontinuierliche Integration und kontinuierliche Bereitstellung.

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 der Ziel-AWS-Service sie empfängt.

Cloud-Kompetenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte Transformationen. Weitere Informationen finden Sie hier: Beiträge von CCoE auf dem AWS-Blog zur Cloud-Unternehmensstrategie.

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 Erstellen, Reifen und Optimieren einer oder mehrerer Cloud-Umgebungen verwendet wird. Weitere Informationen finden Sie unter Erstellen Ihres Cloud-Betriebsmodells.

Phasen der Einführung der Cloud

Die vier Phasen, die Organisationen normalerweise durchlaufen, wenn sie auf die AWS-Cloud migrieren:

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

Diese Phasen wurden von Stephen Orban im Blogbeitrag Der Weg zu Cloud-First und die Phasen der Einführung im AWS-Blog zur Cloud-Unternehmensstrategie definiert. Informationen darüber, wie sie sich auf die AWS-Migrationsstrategie beziehen finden Sie im Leitfaden zur Vorbereitung der Migration.

CMDB

Siehe Konfigurationsverwaltungsdatenbank .

Code-Repository

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

Kalter Cache

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

Kalte Daten

Daten, auf die selten zugegriffen wird und die in der Regel historisch sind. Wenn Sie diese Art von Daten abfragen, sind langsame Abfragen in der Regel akzeptabel. Das Verschieben dieser Daten in Speicherebenen oder -klassen mit geringerer Leistung und geringeren Kosten kann die Kosten senken.

Computer Vision

Ein KI-Bereich, der von Maschinen verwendet wird, um Personen, Orte und Objekte in Bildern mit Genauigkeit auf oder über menschlichen Ebenen zu identifizieren. Oft mit Deep-Learning-Modellen erstellt, automatisiert es die Extraktion, Analyse, Klassifizierung und das Verständnis nützlicher Informationen aus einem einzelnen Bild oder einer Abfolge von Bildern.

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 Compliance- und Sicherheitsprüfungen individuell anzupassen. Mit Hilfe einer YAML-Vorlage können Sie ein Konformitätspaket als einzelne Einheit in einem AWS-Konto und Region oder in einer gesamten Organisation bereitstellen. Weitere Informationen finden Sie unter Konformitätspakete in der AWS Config-Dokumentation.

Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

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

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.

Datenabweichung

Eine bedeutende Variation zwischen den Produktionsdaten und den Daten, die zum Trainieren eines ML-Modells verwendet wurden, oder eine bedeutende Änderung der Eingabedaten im Laufe der Zeit. Datenabweichungen können die Gesamtqualität, Genauigkeit und Fairness bei ML-Modellvorhersagen reduzieren.

Daten während der Übertragung

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

Datenminimierung

Das Prinzip, nur die Daten zu erfassen und zu verarbeiten, die unbedingt erforderlich sind. Durch die Durchführung der Datenminimierung in der AWS Cloud können Datenschutzrisiken, Kosten und Ihren CO2-Fußabdruck für Analysen reduziert werden.

Datenumfang

Eine Reihe präventiver Integritätsschutz in Ihrer -AWSUmgebung, die dazu beitragen, sicherzustellen, dass nur vertrauenswürdige Identitäten von erwarteten Netzwerken aus auf vertrauenswürdige Ressourcen zugreifen. Weitere Informationen finden Sie unter Erstellen 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.

-Datenübereinstimmung

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

Betreff

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

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

Datenbankdefinitionssprache (DDL)

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

Datenbankmanipulationssprache (DML)

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

DDL

Siehe Datenbankdefinitionssprache .

Deep-Ensemble

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

Deep Learning

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

defense-in-depth

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

delegierter Administrator

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

Bereitstellung

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

Entwicklungsumgebung

Siehe Umgebung .

Detektivische Kontrolle

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

Development Value Stream Mapping (DVSM)

Ein Prozess, der verwendet wird, um Einschränkungen zu identifizieren und zu priorisieren, die sich negativ auf Geschwindigkeit und Qualität in einem Lebenszyklus der Softwareentwicklung auswirken. DVSM erweitert den Prozess zur Zuordnung von Wertströmen, der ursprünglich für strenge Herstellungspraktiken entwickelt wurde. Es konzentriert sich auf die Schritte und Teams, die erforderlich sind, um den Wert im Softwareentwicklungsprozess zu schaffen und zu verschieben.

Digitale Telefonie

Eine virtuelle Darstellung eines realen Systems, z. B. eines Gebäudes, einer Fabrik, Industrieanlagen oder einer Produktionslinie. Digitale Komponenten unterstützen prädiktive Wartung, Remote-Überwachung und Produktionsoptimierung.

Dimensionstabelle

In einem Sternschema ist dies eine kleinere Tabelle, die Datenattribute zu quantitativen Daten in einer Faktentabelle enthält. Dimensionstabellenattribute sind in der Regel Textfelder oder diskrete Zahlen, die sich wie Text verhalten. Diese Attribute werden häufig für die Einschränkung, Filterung und Kennzeichnung von Ergebnissätzen verwendet.

Notfall

Ein Ereignis, das verhindert, dass ein Workload oder System seine Geschäftsziele an seinem primär bereitgestellten Standort erfüllt. Bei diesen Ereignissen kann es sich um Naturkatastrophen, technische Ausfälle oder das Ergebnis menschlicher Aktionen wie unbeabsichtigte Fehlkonfigurationen oder Malware-Angriffe handeln.

Notfallwiederherstellung (DR)

Die Strategie und der Prozess, die Sie verwenden, um Ausfallzeiten und Datenverluste zu minimieren, die durch einen Notfall verursacht werden. Weitere Informationen finden Sie unter Notfallwiederherstellung von Workloads auf AWS: Wiederherstellung in der Cloud im AWS Well-Architected Framework.

DML

Siehe Datenbankmanipulationssprache .

Domainorientiertes Design

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

Komponente dient. Dieses Konzept wurde von Eric Evans in seinem Buch Domaingesteuertes Design: Bewältigen der Komplexität im Herzen der Software (Boston: Addison-Wesley Professional, 2003) vorgestellt. Informationen darüber, wie Sie domaingesteuertes Design mit dem Strangler-Fig-Muster verwenden können, finden Sie unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

DR

Siehe Notfallwiederherstellung .

Abweichungserkennung

Nachverfolgen von Abweichungen von einer Basiskonfiguration. Sie können beispielsweise verwenden, AWS CloudFormation um Abweichungen in Systemressourcen zu erkennen, oder Sie können verwenden, AWS Control Tower um Änderungen in Ihrer Landing Zone zu erkennen, die sich auf die Einhaltung der Governance-Anforderungen auswirken könnten.

DVSM

Weitere Informationen finden Sie unter Stream-Zuweisung von Entwicklungswerten.

E

EDA

Weitere Informationen finden Sie unter Explorative Datenanalyse.

Edge-Computing

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

Verschlüsselung

Ein Datenverarbeitungsprozess, der Klartextdaten, die für Menschen lesbar sind, in Geheimtext 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 Endpunktservice mit AWS PrivateLink erstellen und anderen AWS-Konten oder AWS Identity and Access Management (IAM)-Prinzipalen Berechtigungen erteilen. 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.

Envelope-Verschlüsselung

Der Prozess der Verschlüsselung eines Verschlüsselungsschlüssels mit einem anderen Verschlüsselungsschlüssel. Weitere Informationen finden Sie unter Umschlagverschlüsselung 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 AWS-CAF-Sicherheitsepics 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.

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 . Es speichert quantitative Daten über den Geschäftsbetrieb. In der Regel enthält eine Faktentabelle zwei Arten von Spalten: Spalten, die Messwerte enthalten, und Spalten, die einen Fremdschlüssel für eine Dimensionstabelle enthalten.

schnell fehlschlagen

Eine Variante, die häufige und inkrementelle Tests verwendet, um den Entwicklungslebenszyklus zu reduzieren. Es ist ein wichtiger Teil eines agilen Ansatzes.

Fehlerisolierungsgrenze

In der AWS Cloudeine Grenze wie eine Availability Zone, , AWS-RegionSteuerebene oder Datenebene, die die Auswirkungen eines Ausfalls begrenzt und die Ausfallsicherheit von Workloads verbessert. Weitere Informationen finden Sie unter AWS Fehlerisolierungsgrenzen.

Feature-Zweig

Siehe Verzweigung .

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 Machine-Learning-Modellen mit :AWS.

Featuretransformation

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

FGAC

Siehe differenzierte Zugriffskontrolle .

differenzierte Zugriffskontrolle (FGAC)

Die Verwendung mehrerer Bedingungen zum Zulassen oder Verweigern einer Zugriffsanforderung.

Flash-Cut-Migration

Eine Datenbankmigrationsmethode, die die kontinuierliche Datenreplikation über die Erfassung von Änderungsdaten verwendet, um Daten in der kürzestmöglichen Zeit zu migrieren, anstatt einen schrittweisen Ansatz zu verwenden. Ziel ist es, Ausfallzeiten auf ein Minimum zu beschränken.

G

Geoblockierung

Siehe geografische Einschränkungen.

Geografische Einschränkungen (Geoblocking)

In Amazon eine Option CloudFront, um Benutzer in bestimmten Ländern am Zugriff auf Inhaltsverteilungen zu hindern. Sie können eine Zulassungsliste oder eine Sperrliste verwenden, um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie unter Einschränken der geografischen Verteilung Ihrer Inhalte in der - CloudFront Dokumentation.

Gitflow-Workflow

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

Greenfield-Strategie

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

Integritätsschutz

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

H

HA

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.

Hochverfügbarkeit (HA)

Die Fähigkeit eines Workloads, im Falle von Herausforderungen oder Katastrophen kontinuierlich und ohne Eingriff zu arbeiten. HA-Systeme sind so konzipiert, dass sie automatisch ein Failover durchführen, konsistent eine qualitativ hochwertige Leistung liefern und verschiedene Lasten und Ausfälle mit minimalen Leistungseinbußen behandeln.

Historische Modernisierung

Ein Ansatz, der zur Modernisierung und Aktualisierung von Systemen der Betriebstechnologie (OT) verwendet wird, um die Anforderungen der Fertigung besser zu erfüllen. Ein Historiker ist eine Art von Datenbank, die verwendet wird, um Daten aus verschiedenen Quellen in einer Fabrik zu sammeln und zu speichern.

Homogene Datenbankmigration

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

Hot Data

Daten, auf die häufig zugegriffen wird, wie Echtzeitdaten oder aktuelle Übersetzungsdaten. Diese Daten erfordern in der Regel eine Speicherebene oder Klasse mit hoher Leistung, um schnelle Abfrageantworten bereitzustellen.

Hotfix

Eine dringende Lösung für ein kritisches Problem in einer Produktionsumgebung. Aufgrund seiner Wichtigkeit 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

Siehe 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

Weitere Informationen finden Sie unter Industrielles Internet der Dinge.

unveränderliche Infrastruktur

Ein Modell, das eine neue Infrastruktur für Produktions-Workloads bereitstellt, anstatt die vorhandene Infrastruktur zu aktualisieren, zu patchen oder zu ändern. Unveränderliche Infrastrukturen sind von Natur aus konsistenter, zuverlässiger und vorhersehbarer als veränderbare Infrastrukturen. Weitere Informationen finden Sie unter Bereitstellung mit unveränderlicher Infrastruktur im AWS Well-Architected Framework.

Eingehende (ingress) VPC

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

Inkrementelle Migration

Eine Cutover-Strategie, bei der Sie Ihre Anwendung in kleinen Teilen migrieren, anstatt eine einziges vollständiges Cutover durchzuführen. Beispielsweise könnten Sie zunächst nur einige Microservices oder Benutzer auf das neue System umstellen. Nachdem Sie sich vergewissert haben, dass alles ordnungsgemäß funktioniert, können Sie weitere Microservices oder Benutzer

schrittweise verschieben, bis Sie Ihr Legacy-System außer Betrieb nehmen können. Diese Strategie reduziert die mit großen Migrationen verbundenen Risiken.

Infrastruktur

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

Infrastructure as Code (IaC)

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

Industrielles Internet der Dinge (IIoT)

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

Inspektions-VPC

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

Internet of Things (IoT)

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

Interpretierbarkeit

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

IoT

Siehe Internet der Dinge.

IT information library (ITIL, IT-Informationsbibliothek)

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

T service management (ITSM, IT-Servicemanagement)

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

ITIL

Siehe IT-Informationsbibliothek .

ITSM

Siehe IT-Servicemanagement.

L

Labelbasierte Zugriffskontrolle (LBAC)

Eine Implementierung der obligatorischen Zugriffskontrolle (MAC), bei der den Benutzern und den Daten selbst jeweils explizit ein Sicherheitsbezeichnungswert zugewiesen wird. Die Schnittmenge zwischen der Benutzersicherheitsbezeichnung und der Datensicherheitsbezeichnung bestimmt, welche Zeilen und Spalten vom Benutzer angezeigt werden können.

Landing Zone

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

Große Migration

Eine Migration von 300 oder mehr Servern.

LBAC

Siehe Label-basierte Zugriffskontrolle .

Geringste Berechtigung

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

Lift and Shift

Siehe 7 Rs.

Little-Endian-System

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

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 Verzweigung .

Von verwaltete Services

AWS-Services für die die Infrastrukturebene, das Betriebssystem und die Plattformen AWS betreibt und Sie auf die Endpunkte zugreifen, um Daten zu speichern und abzurufen. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB sind Beispiele für verwaltete Services. Diese werden auch als abstrakte Services bezeichnet.

MAP

Weitere Informationen finden Sie unter Migration Acceleration Program.

Mechanismus

Ein vollständiger Prozess, in dem Sie ein Tool erstellen, die Einführung des Tools fördern und dann die Ergebnisse überprüfen, um Anpassungen vorzunehmen. Ein Mechanismus ist ein

Zyklus, der sich bei der Ausführung selbst verstärkt und verbessert. Weitere Informationen finden Sie unter Erstellen von Mechanismen im AWS Well-Architected Framework.

Mitgliedskonto

Alle AWS-Konten mit Ausnahme des Verwaltungskontos, die Teil einer Organisation in AWS Organizations sind. Ein Konto kann jeweils nur einer Organisation angehören.

Microservice

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

Microservices-Architekturen

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

Migration Acceleration Program (MAP)

Ein AWS-Programm, das Beratung, Unterstützung, Training und Services bietet, um Organisationen dabei zu unterstützen, eine solide betriebliche Grundlage für den Umstieg 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 der Migrationsfabrik gehören in der Regel Betrieb, 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 das AWS-Konto.

Migrationsmuster

Eine wiederholbare Migrationsaufgabe, in der die Migrationsstrategie, das Migrationsziel und die verwendete Migrationsanwendung oder der verwendete Migrationsservice detailliert beschrieben werden. Beispiel: Hostwechsel-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 in die AWS-Cloud. MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung, Migrationspriorisierung und Wellenplanung). Das MPA-Tool (erfordert Anmeldung) ist für alle AWS-Berater und APN-Partnerberater kostenlos verfügbar.

Migration Readiness Assessment (MRA)

Der Prozess der Gewinnung von Erkenntnissen über die Cloud-Bereitschaft einer Organisation, der Identifizierung von Stärken und Schwächen und der Erstellung eines Aktionsplans zur Schließung identifizierter Lücken unter Verwendung des AWS-CAF. Weitere Informationen finden Sie im Benutzerhandbuch für Migration Readiness. MRA ist die erste Phase der AWS-Migrationsstrategie.

Migrationsstrategie

Der Ansatz, der verwendet wird, um einen Workload in die AWS-Cloud zu migrieren. Weitere Informationen finden Sie im Eintrag 7 Rs in diesem Glossar und unter Mobilisieren Ihrer Organisation zur Beschleunigung umfangreicher Migrationen.

ML

Siehe Machine Learning.

MPA

Weitere Informationen finden Sie unter Bewertung des Migrationsportfolios.

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 Bewertung 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.

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?“

veränderliche Infrastruktur

Ein Modell, das die vorhandene Infrastruktur für Produktions-Workloads aktualisiert und ändert. Um Konsistenz, Zuverlässigkeit und Vorhersehbarkeit zu verbessern, empfiehlt das AWS Well-Architected Framework die Verwendung unveränderlicher Infrastruktur als bewährte Methode.

O

OAC

Siehe Ursprungszugriffskontrolle .

OAI

Siehe Ursprungszugriffsidentität .

COM

Siehe Organisations-Änderungsmanagement.

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 Betriebsebene .

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.

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, den Umfang von Vorfällen und möglichen Ausfällen zu verstehen, zu bewerten, zu verhindern oder zu reduzieren. Weitere Informationen finden Sie unter Operational Readiness Reviews (ORR) im AWS Well-Architected Framework.

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

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

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 heißt dieser Rahmen Beschleunigung der Menschen, aufgrund der Geschwindigkeit des Wandels, der bei Projekten zur Cloud-Einführung erforderlich ist. Weitere Informationen finden Sie im OCM-Handbuch.

Ursprungszugriffskontrolle (OAC)

In CloudFront, eine erweiterte Option zur Einschränkung des Zugriffs auf die Sicherung Ihrer Amazon Simple Storage Service (Amazon S3)-Inhalte. OAC unterstützt alle S3-Buckets in allen AWS-Regionen, serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) und dynamische `PUT`- und `DELETE`-Anforderungen an den S3-Bucket.

Ursprungszugriffsidentität (OAI)

In CloudFront, eine Option zur Einschränkung des Zugriffs auf die Sicherung Ihrer Amazon S3-Inhalte. Wenn Sie OAI verwenden, CloudFront erstellt einen Prinzipal, mit dem sich Amazon S3 authentifizieren kann. Authentifizierte Prinzipale können nur über eine bestimmte CloudFront Verteilung auf Inhalte in einem S3-Bucket zugreifen. Siehe auch OAC, das eine detailliertere und verbesserte Zugriffskontrolle bietet.

ORR

Siehe Überprüfung der Betriebsbereitschaft.

Ausgehende (egress) VPC

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

P

Berechtigungsgrenze

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

persönlich identifizierbare Informationen (PII)

Informationen, die bei direkter Anzeige oder in Verbindung mit anderen zugehörigen Daten verwendet werden können, um die Identität einer Person verfolgbar abzuleiten. Beispiele für PII 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.

policy

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

Polyglotte Beharrlichkeit

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

Portfoliobewertung

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

predicate

Eine Abfragebedingung, die `true` oder zurückgibt`false`, die sich üblicherweise in einer -`WHERE`Klausel befindet.

Prädikat-Pushdown

Eine Datenbankabfrageoptimierungstechnik, die die Daten in der Abfrage vor der Übertragung filtert. Dies reduziert die Datenmenge, die aus der relationalen Datenbank abgerufen und verarbeitet werden muss, und verbessert die Abfrageleistung.

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 in AWS, die Aktionen durchführen und auf Ressourcen zugreifen kann. Diese Entität ist normalerweise ein Root-Benutzer für ein AWS-Konto, eine IAM-Rolle oder ein Benutzer.

Weitere Informationen finden Sie unter Prinzipal in Rollenbegriffe und -konzepte in der IAM-Dokumentation.

Datenschutz nach Design

Ein Ansatz im System-Engineering, der den Datenschutz während des gesamten Entwicklungsprozesses berücksichtigt.

Privat gehostete Zonen

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

Proaktive Kontrolle

Eine Sicherheitskontrolle, die die Bereitstellung nicht konformer Ressourcen verhindert. Diese Kontrollen scannen Ressourcen, bevor sie bereitgestellt werden. Wenn die Ressource nicht mit der Kontrolle kompatibel ist, wird sie nicht bereitgestellt. Weitere Informationen finden Sie im Referenzhandbuch zu Kontrollen in der -AWS Control TowerDokumentation und unter Proaktive Kontrollen in Implementierung von Sicherheitskontrollen in AWS.

Produktionsumgebung

Siehe Umgebung .

Visualisierung

Der Prozess zum Ersetzen persönlicher Kennungen in einem Datensatz durch Platzhalterwerte. Die Pseudonymisierung kann dazu beitragen, den persönlichen Datenschutz zu schützen. Pseudonymisierte Daten werden weiterhin als personenbezogene Daten betrachtet.

Q

Abfrageplan

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

Abfrageplanregression

Wenn ein Datenbankserviceoptimierer einen weniger optimalen Plan wählt als vor einer bestimmten Änderung der Datenbankumgebung. Dies kann durch Änderungen an Statistiken,

Beschränkungen, Umgebungseinstellungen, Abfrageparameter-Bindungen und Aktualisierungen der Datenbank-Engine verursacht werden.

R

RACI-Matrix

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

Ransomware

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

RASCI-Matrix

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

RCAC

Siehe Zugriffskontrolle für Zeilen und Spalten .

Read Replica

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

Neuarchitektur

Siehe 7 Rs.

Recovery Point Objective (RPO)

Die maximal zulässige Zeit seit dem letzten Datenwiederherstellungspunkt. Dies bestimmt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Unterbrechung des Services angesehen wird.

Recovery Time Objective (RTO)

Die maximal akzeptable Verzögerung zwischen der Unterbrechung des Services und der Wiederherstellung des Services.

Faktorwechsel

Siehe 7 Rs.

Region

Eine Sammlung von AWS-Ressourcen in einem geografischen Bereich. Jede AWS-Region ist isoliert und unabhängig von den anderen, um Fehlertoleranz, Stabilität und Belastbarkeit zu gewährleisten. Weitere Informationen finden Sie unter Verwalten von AWS-Regionen in der Allgemeine AWS-Referenz.

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.

Hostwechsel

Siehe 7 Rs.

Veröffentlichung

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

Verschiebung

Siehe 7 Rs.

Plattformwechsel

Siehe 7 Rs.

Neukauf

Siehe 7 Rs.

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 wird aus den in der Matrix definierten

Verantwortungstypen abgeleitet: verantwortlich (R), rechenschaftspflichtig (A), konsultiert (C) und informiert (I). Der Support-Typ (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.

Außerbetriebnahme

Siehe 7 Rs.

Drehung

Der Prozess der regelmäßigen Aktualisierung eines Secrets, um es einem Angreifer zu erschweren, auf die Anmeldeinformationen zuzugreifen.

Zeilen- und Spaltenzugriffskontrolle (RCAC)

Die Verwendung grundlegender, flexibler SQL-Ausdrücke, für die Zugriffsregeln definiert sind. RCAC besteht aus Zeilenberechtigungen und Spaltenmasken.

RPO

Siehe Recovery Point Objective .

RTO

Siehe Recovery Time Objective .

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. Dieses Feature ermöglicht verbundenes Single Sign-On (SSO), sodass Benutzer sich bei der AWS Management Console anmelden oder die AWS-API-Operationen aufrufen können, ohne dass Sie in IAM einen Benutzer für jeden 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.

SCP

Siehe Service-Kontrollrichtlinie .

Secret

In vertrauliche AWS Secrets Manageroder eingeschränkte Informationen, wie z. B. ein Passwort oder Benutzeranmeldeinformationen, die Sie in verschlüsselter Form speichern. Sie besteht aus dem Secret-Wert und seinen Metadaten. Der Secret-Wert kann binär, eine einzelne Zeichenfolge oder mehrere Zeichenfolgen sein. Weitere Informationen finden Sie unter Secret in der Secrets-Manager-Dokumentation.

Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: präventive , detektivische , reaktive und proaktive .

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 der Sicherheitsantwort

Eine vordefinierte und geprogrammete Aktion, die darauf ausgelegt ist, automatisch auf ein Sicherheitsereignis zu reagieren oder es zu beheben. Diese Automatisierungen dienen als detektivische oder reaktive 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-Instance oder das Rotieren von Anmeldeinformationen.

Serverseitige Verschlüsselung

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

Service-Kontrollrichtlinie (SCP)

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

Service-Endpunkt

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

Service Level Agreement (SLA)

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

Indikator auf Serviceebene (SLI)

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

Service Level Objective (SLO)

Eine Zielmetrik, die den Zustand eines Services darstellt, gemessen durch einen Indikator auf Serviceebene .

Modell der geteilten Verantwortung

Ein Modell, das die Verantwortung beschreibt, die Sie mit AWS für die Sicherheit der Cloud und die Einhaltung der Vorschriften teilen. AWS ist für die Sicherheit der Cloud zuständig, während Sie für die Sicherheit in der Cloud verantwortlich sind. Weitere Informationen finden Sie unter Modell der geteilten Verantwortung.

SIEM

Siehe Sicherheitsinformationen und Ereignisverwaltungssystem .

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 Indikator auf Serviceebene .

SLO

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 im AWS Cloud.

SPOF

Siehe einzelne Fehlerquelle .

Sternschema

Eine Datenbankorganisationsstruktur, 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.

Symmetrische Verschlüsselung

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

synthetische Tests

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

T

tags

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

Zielvariable

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

Aufgabenliste

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

Testumgebungen

Siehe Umgebung .

Training

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

Transit-Gateway

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

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

Erteilen von Berechtigungen für einen Service, den Sie für die Ausführung von Aufgaben in AWS Organizations und in ihren Konten und in Ihrem Namen in Ihrer Organisation angeben. 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 unter Verwenden von AWS Organizations mit anderen AWS-Services in der AWS Organizations-Dokumentation.

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 Pzzas 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

Auch bekannt als „schwere Arbeit“, eine Arbeit, die erforderlich ist, um eine Anwendung zu erstellen und zu betreiben, aber die dem Endbenutzer keinen direkten Wert bietet oder einen kompetitiven Vorteil bietet. Beispiele für undifferenzierte Aufgaben sind Beschaffung, Wartung und Kapazitätsplanung.

höhere Umgebungen

Siehe Umgebung .

V

Vacuuming

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

Versionskontrolle

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

VPC-Peering

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

Schwachstelle

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

Warm-Daten

Daten, auf die selten zugegriffen wird. Bei Abfragen 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, z. B. die Berechnung eines gleitenden Durchschnitts oder den Zugriff auf den Wert von Zeilen basierend auf 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.

WORM

Siehe einmal schreiben, viele lesen.

WQF

Weitere Informationen finden Sie unter AWS Workload Qualification Framework.

einmal schreiben, viele lesen (WORM)

Ein Speichermodell, das Daten einmalig 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-Exploits

Ein Angriff, in der Regel Malware, der eine Zero-Day-Schwachstelle nutzt.

Zero-Day-Schwachstelle

Ein nicht behobener Fehler oder eine Schwachstelle in einem Produktionssystem. Bedrohungsakteure können diese Art von Schwachstelle verwenden, um das System anzugreifen. Entwickler werden häufig aufgrund des Angriffs auf die Schwachstelle aufmerksam.

Zombie-Anwendung

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

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.