



Modellieren von Daten mit Amazon DynamoDB

# AWS Präskriptive Leitlinien



# AWS Präskriptive Leitlinien: Modellieren von Daten mit Amazon DynamoDB

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

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

# Table of Contents

Einführung .....	1
Prozessablauf .....	2
RACI-Matrix .....	2
Prozessschritte .....	5
Schritt 1. Identifizieren der Anwendungsfälle und des logischen Datenmodells .....	5
Ziele .....	5
Prozess .....	5
Tools und Ressourcen .....	6
RACI .....	6
Outputs .....	6
Schritt 2. Erstellen einer vorläufigen Kostenschätzung .....	6
Ziel .....	6
Prozess .....	6
Tools und Ressourcen .....	7
RACI .....	7
Outputs .....	7
Schritt 3. Identifizieren Ihrer Datenzugriffsmuster .....	7
Ziel .....	7
Prozess .....	7
Tools und Ressourcen .....	8
RACI .....	9
Outputs .....	9
Beispiel .....	9
Schritt 4. Identifizieren der technischen Anforderungen .....	10
Ziel .....	10
Prozess .....	10
Tools und Ressourcen .....	10
RACI .....	10
Outputs .....	10
Schritt 5. Erstellen des DynamoDB-Datenmodells .....	10
Ziel .....	10
Prozess .....	11
Tools und Ressourcen .....	12
RACI .....	12

Outputs .....	12
Beispiel .....	13
Schritt 6: Erstellen der Datenabfragen .....	13
Ziel .....	13
Prozess .....	13
Tools und Ressourcen .....	14
RACI .....	14
Outputs .....	14
Beispiele .....	14
Schritt 7. Validieren des Datenmodells .....	15
Ziel .....	15
Prozess .....	15
Tools und Ressourcen .....	15
RACI .....	15
Outputs .....	15
Schritt 8. Überprüfen der Kostenschätzung .....	16
Ziele .....	16
Prozess .....	16
Tools und Ressourcen .....	16
RACI .....	16
Outputs .....	17
Schritt 9. Bereitstellen des Datenmodells .....	17
Ziel .....	17
Prozess .....	17
Tools und Ressourcen .....	17
RACI .....	17
Outputs .....	17
Beispiel .....	18
Vorlagen .....	19
Bewertungsvorlage für Geschäftsanforderungen .....	19
Bewertungsvorlage für technische Anforderungen .....	23
Vorlage für Zugriffsmuster .....	27
Vorlage .....	28
Bewährte Methoden .....	34
Hierarchische Datenmodellierung .....	35
Schritt 1: Identifizieren der Anwendungsfälle und des logischen Datenmodells .....	35

Schritt 2: Erstellen einer vorläufigen Kostenschätzung .....	38
Schritt 3: Identifizieren Ihrer Datenzugriffsmuster .....	38
Schritt 4: Identifizieren der technischen Anforderungen .....	39
Schritt 5: Erstellen eines DynamoDB-Datenmodells .....	40
Speichern von Komponenten in der Tabelle .....	40
Der GSI1-Index .....	42
Der GSI2-Index .....	42
Schritt 6: Erstellen von Datenabfragen .....	43
Schritt 7: Validieren des Datenmodells .....	47
Schritt 8: Überprüfen der Kostenschätzung .....	48
Ziele .....	48
Prozess .....	48
Schritt 9: Bereitstellen des Datenmodells .....	49
Weitere Ressourcen .....	51
Mitwirkende .....	53
Dokumentverlauf .....	54
Glossar .....	55
# .....	55
A .....	56
B .....	59
C .....	61
D .....	64
E .....	69
F .....	71
G .....	72
H .....	73
I .....	74
L .....	77
M .....	78
O .....	82
P .....	85
Q .....	88
R .....	88
S .....	91
T .....	95
U .....	97

---

V .....	97
W .....	98
Z .....	99
.....	c

# Modellieren von Daten mit Amazon DynamoDB

## Prozess, Vorlagen und bewährte Methoden

Amazon Web Services ([Mitwirkende](#))

Dezember 2023 ([Dokumentverlauf](#))

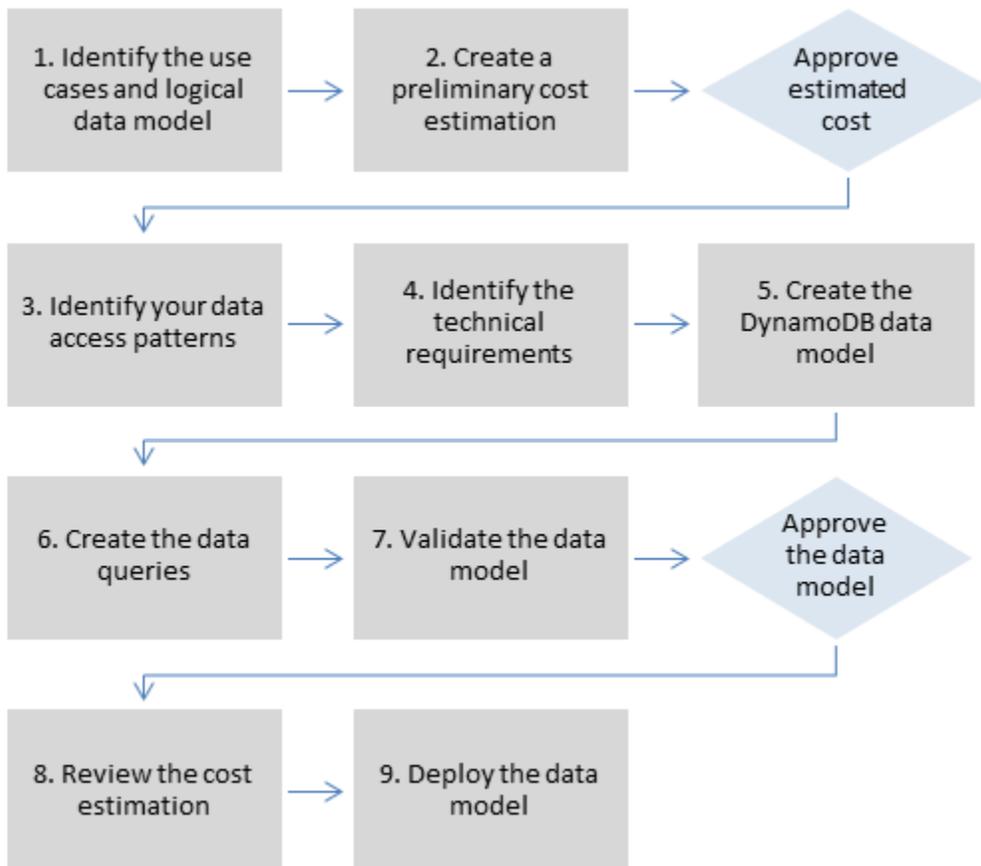
Keine SQL Datenbank bietet flexible Schemas für die Erstellung moderner Anwendungen. Sie sind weithin bekannt für ihre einfache Entwicklung, Funktionalität und Skalierbarkeit. Amazon DynamoDB bietet schnelle und vorhersehbare Leistung mit nahtloser Skalierbarkeit für SQL Nox-Datenbanken in der Amazon Web Services (AWS) Cloud. Als vollständig verwalteter Datenbankdienst hilft Ihnen DynamoDB dabei, den administrativen Aufwand für den Betrieb und die Skalierung einer verteilten Datenbank zu verringern. Sie müssen sich keine Gedanken über Hardwarebereitstellung, Einrichtung und Konfiguration, Replikation, Software-Patches oder Clusterskalierung machen.

Kein SQL Schemadesign erfordert einen anderen Ansatz als das herkömmliche Design eines relationalen Datenbankmanagementsystems (RDBMS). RDBMS Ein Datenmodell konzentriert sich auf die Struktur von Daten und ihre Beziehungen zu anderen Daten. Keine SQL Datenmodellierung konzentriert sich auf Zugriffsmuster oder darauf, wie die Anwendung die Daten nutzen wird. Deshalb werden die Daten so gespeichert, dass einfache Abfrageoperationen unterstützt werden. Für ein RDBMS System wie Microsoft SQL Server oder IBM Db2 können Sie ein normalisiertes Datenmodell erstellen, ohne viel über Zugriffsmuster nachdenken zu müssen. Sie können das Datenmodell erweitern, um Ihre Muster und Abfragen später zu unterstützen.

In diesem Handbuch wird ein Datenmodellierungsprozess für die Verwendung von DynamoDB vorgestellt, der funktionale Anforderungen, Leistung und effektive Kosten enthält. Das Handbuch richtet sich an Datenbanktechniker, die planen, DynamoDB als Betriebsdatenbank für ihre Anwendungen zu verwenden, auf denen sie ausgeführt werden. AWS AWS Professional Services hat das empfohlene Verfahren verwendet, um Unternehmen bei der DynamoDB-Datenmodellierung für verschiedene Anwendungsfälle und Workloads zu unterstützen.

# Ablauf des Datenmodellierungsprozesses

Wir empfehlen den folgenden Prozess, wenn Sie Daten mit Amazon DynamoDB modellieren. Die Schritte werden [später in diesem Leitfaden](#) ausführlich erörtert.



## RACI-Matrix

Einige Organisationen verwenden eine Matrix zur Zuweisung von Verantwortlichkeiten (auch RACI-Matrix genannt), um die verschiedenen Rollen zu beschreiben, die an einem bestimmten Projekt oder Geschäftsbetrieb beteiligt sind. In diesem Leitfaden wird eine vorgeschlagene RACI-Matrix vorgestellt, die Ihrer Organisation helfen könnte, die richtigen Personen und richtigen Verantwortlichkeiten für den DynamoDB-Datenmodellierungsprozess zu identifizieren. Für jeden Schritt des Prozesses werden die Stakeholder und ihre Beteiligung aufgeführt:

- R – verantwortlich für den Abschluss des Schritts

- A – verantwortlich für die Genehmigung und Abmeldung der Arbeit
- C – konsultiert, um Eingaben für eine Aufgabe bereitzustellen
- I – über den Fortschritt informiert, aber nicht direkt an der Aufgabe beteiligt

Abhängig von der Struktur Ihrer Organisation und Ihres Projektteams können die Rollen in der folgenden RACI-Matrix von demselben Stakeholder ausgeführt werden. In einigen Situationen sind die Stakeholder sowohl für bestimmte Schritte verantwortlich als auch dafür verantwortlich. Datenbankingenieure können beispielsweise sowohl für die Erstellung als auch für die Genehmigung des Datenmodells verantwortlich sein, da dies ihr Domainbereich ist.

Prozessschritt	Geschäftsbenu- tzer	Geschäfts- analyst	Lösungsar- chitekt	Datenbank- ingenieur	Anwendung- sentwickler	DevOps Techniker
1. Identifizieren der Anwendungsfälle und des logischen Datenmodells	C	R/A	I	R		
2. Erstellen einer vorläufigen Kostenschätzung	C	A	I	R		
3. Identifizieren Ihrer Datenzugriffsmuster	C	A	I	R		
4. Identifizieren der technischen	C	C	A	R		

Prozessschritt	Geschäftsbenuer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
Anforderungen						
5. Erstellen des DynamoDB-Datenmodells	I	I	I	R/A		
6. Erstellen der Datenabfragen	I	I	I	R/A	R	
7. Validieren des Datenmodells	A	R	I	C		
8. Überprüfen der Kostenschätzung	C	A	I	R		
9. Bereitstellen des DynamoDB-Datenmodells	I	I	C	C		R/A

# Schritte des Datenmodellierungsprozesses

In diesem Abschnitt werden die einzelnen Schritte des empfohlenen Datenmodellierungsprozesses für Amazon DynamoDB beschrieben.

## Themen

- [Schritt 1. Identifizieren der Anwendungsfälle und des logischen Datenmodells](#)
- [Schritt 2. Erstellen einer vorläufigen Kostenschätzung](#)
- [Schritt 3. Identifizieren Ihrer Datenzugriffsmuster](#)
- [Schritt 4. Identifizieren der technischen Anforderungen](#)
- [Schritt 5. Erstellen des DynamoDB-Datenmodells](#)
- [Schritt 6: Erstellen der Datenabfragen](#)
- [Schritt 7. Validieren des Datenmodells](#)
- [Schritt 8. Überprüfen der Kostenschätzung](#)
- [Schritt 9. Bereitstellen des Datenmodells](#)

## Schritt 1. Identifizieren der Anwendungsfälle und des logischen Datenmodells

### Ziele

- Erfassen Sie die Geschäftsanforderungen und Anwendungsfälle, die eine NoSQL-Datenbank erfordern.
- Definieren Sie das logische Datenmodell mithilfe eines Entity-Relationship (ER)-Diagramms.

### Prozess

- Geschäftsanalysten interviewen Geschäftsanwender, um die Anwendungsfälle und die erwarteten Ergebnisse zu identifizieren.
- Der Datenbankingenieur erstellt das konzeptionelle Datenmodell.
- Der Datenbankingenieur erstellt das logische Datenmodell.

- Der Datenbankingenieur sammelt Informationen über Elementgröße, Datenvolumen und den erwarteten Lese- und Schreibdurchsatz.

## Tools und Ressourcen

- Bewertung der Geschäftsanforderungen (siehe [Vorlage](#) )
- Zugriffsmustermatrix (siehe [Vorlage](#) )
- Ihr bevorzugtes Tool zum Erstellen von Diagrammen

## RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
C	R/A	I	R		

## Outputs

- Dokumentierte Anwendungsfälle und Geschäftsanforderungen
- Logisches Datenmodell (ER-Diagramm)

## Schritt 2. Erstellen einer vorläufigen Kostenschätzung

### Ziel

- Entwickeln Sie eine vorläufige Kostenschätzung für DynamoDB .

### Prozess

- Der Datenbankingenieur erstellt die erste Kostenanalyse anhand verfügbarer Informationen und der Beispiele, die auf der [DynamoDB-Preisseite](#) dargestellt werden.
  - Erstellen Sie eine Kostenschätzung für On-Demand-Kapazität (siehe [Beispiel](#) ).
  - Erstellen Sie eine Kostenschätzung für bereitgestellte Kapazität (siehe [Beispiel](#) ).

- Rufen Sie für das Modell der bereitgestellten Kapazität die Kosten für die Schätzung vom Rechner ab und wenden Sie den Rabatt für reservierte Kapazität an.
- Vergleichen Sie die geschätzten Kosten der beiden Kapazitätsmodelle.
- Erstellen Sie eine Schätzung für alle Umgebungen (Dev, Prod, QA).
- Der Geschäftsanalyst überprüft und genehmigt oder lehnt die vorläufige Kostenschätzung ab.

## Tools und Ressourcen

- [AWS Pricing Calculator](#)

## RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
C	A	I	R		

## Outputs

- Vorläufige Kostenschätzung

## Schritt 3. Identifizieren Ihrer Datenzugriffsmuster

Zugriffsmuster oder Abfragemuster definieren, wie die Benutzer und das System auf die Daten zugreifen, um die Geschäftsanforderungen zu erfüllen.

### Ziel

- Dokumentieren Sie die Datenzugriffsmuster.

### Prozess

- Datenbankingenieur und Geschäftsanalyst stellen die Endbenutzer vor, um zu ermitteln, wie Daten mithilfe der Matrixvorlage für Datenzugriffsmuster abgefragt werden.

- Bei neuen Anwendungen überprüfen sie die Erfahrungen der Benutzer über Aktivitäten und Ziele. Sie dokumentieren die Anwendungsfälle und analysieren die Zugriffsmuster, die für die Anwendungsfälle erforderlich sind.
- Für vorhandene Anwendungen analysieren sie Abfrageprotokolle, um herauszufinden, wie Menschen das System derzeit verwenden, und um die wichtigsten Zugriffsmuster zu identifizieren.
- Der Datenbankingenieur identifiziert die folgenden Eigenschaften der Zugriffsmuster:
  - Datengröße: Wenn Sie wissen, wie viele Daten gleichzeitig gespeichert und angefordert werden, können Sie ermitteln, wie die Daten am effektivsten partitioniert werden können (siehe [Blogbeitrag](#)).
  - Datenform: Statt Daten neu zu gestalten, wenn eine Abfrage verarbeitet wird (wie im Fall von RDBMS-Systemen), organisieren NoSQL-Datenbanken Daten so, dass ihre Form in der Datenbank dem entspricht, was abgefragt werden wird. Dies ist ein entscheidender Faktor, um Geschwindigkeit und Skalierbarkeit zu verbessern.
  - Datengeschwindigkeit: DynamoDB wird durch die Erhöhung der Anzahl der physischen Partitionen skaliert, die für die Verarbeitung von Abfragen verfügbar sind, sowie durch die effiziente Verteilung von Daten über diese Partitionen. Wenn Sie die Spitzenlast von Abfragen im Voraus kennen, können Sie bestimmen, wie Daten partitioniert werden, um die E/A-Kapazität optimal zu nutzen.
- Der Geschäftsbenutzer priorisiert die Zugriffs- oder Abfragemuster.
  - Prioritätsabfragen sind in der Regel die am häufigsten verwendeten oder relevantesten Abfragen. Es ist auch wichtig, Abfragen zu identifizieren, die eine geringere Antwortlatenz erfordern.

## Tools und Ressourcen

- Zugriffsmustermatrix (siehe [Vorlage](#))
- [Auswahl des richtigen DynamoDB-Partitionsschlüssels](#) (AWS-Datenbank-Blog)
- [NoSQL-Design für DynamoDB](#) (DynamoDB-Dokumentation)

## RACI

Geschäftsb benutzer	Geschäfts analyst	Lösungsar chitekt	Datenbank ingenieur	Anwendung sentwickler	DevOps Techniker
C	A	I	R		

## Outputs

- Matrix für Datenzugriffsmuster

## Beispiel

Zugriffsm uster	Priorität	Lesen oder Schreiben	Beschreib ung	Typ (Einzelel ement, mehrere Elemente oder alle)	Schlüssel attribut	Filter	Ergebnisr eihenfolg e
Benutzerp rofil erstellen	Hoch	Schreiben	Der Benutzer erstellt ein neues Profil	Einzelner Artikel	Username	N/A	N/A
Benutzerp rofil aktualisi eren	Mittelsch wer	Schreiben	Benutzer aktualisi ert sein Profil	Einzelner Artikel	Username	Benutzern ame = aktueller Benutzer	N/A

## Schritt 4. Identifizieren der technischen Anforderungen

### Ziel

- Erfassen Sie die technischen Anforderungen für die DynamoDB-Datenbank.

### Prozess

- Geschäftsanalysten fragen den Geschäftsbutzer und das DevOps Team ab, um die technischen Anforderungen mithilfe des Bewertungsprüfers zu ermitteln.

### Tools und Ressourcen

- Bewertung der technischen Anforderungen (siehe [Beispiel-Artikel](#))

### RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
C	C	A	R		

### Outputs

- Dokument zu den technischen Anforderungen

## Schritt 5. Erstellen des DynamoDB-Datenmodells

### Ziel

- Erstellen Sie das DynamoDB-Datenmodell.

## Prozess

- Der Datenbankingenieur gibt an, wie viele Tabellen für jeden Anwendungsfall erforderlich sind. Wir empfehlen, so wenige Tabellen wie möglich in einer DynamoDB-Anwendung zu verwalten.
- Identifizieren Sie auf der Grundlage der häufigsten Zugriffsmuster den Primärschlüssel, der einer von zwei Typen sein kann: ein Primärschlüssel mit einem Partitionsschlüssel, der Daten identifiziert, oder ein Primärschlüssel mit einem Partitionsschlüssel und einem Sortierschlüssel. Ein Sortierschlüssel ist ein sekundärer Schlüssel zum Gruppieren und Organisieren von Daten, damit sie innerhalb einer Partition effizient abgefragt werden können. Sie können Sortierschlüssel verwenden, um hierarchische Beziehungen in Ihren Daten zu definieren, die Sie auf jeder Ebene der Hierarchie abfragen können (siehe [Blogbeitrag](#) ).
- Entwerfen von Partitionsschlüsseln
  - Definieren Sie den Partitionsschlüssel und werten Sie dessen Verteilung aus.
  - Identifizieren Sie die Notwendigkeit von [Schreib-Sharding](#), um Workloads gleichmäßig zu verteilen.
- Sortierschlüsselentwurf
  - Identifizieren Sie den Sortierschlüssel.
  - Identifizieren Sie die Notwendigkeit eines zusammengesetzten Sortierschlüssels.
  - Identifizieren Sie die Notwendigkeit einer Versionskontrolle.
- Identifizieren Sie auf der Grundlage der Zugriffsmuster die sekundären Indizes, um die Abfrageanforderungen zu erfüllen.
- Identifizieren Sie die Notwendigkeit [lokaler sekundärer Indizes](#) (LSIs). Dies sind Indizes, die denselben Partitionsschlüssel wie die Basistabelle haben, aber einen anderen Sortierschlüssel.
  - Für Tabellen mit LSIs gibt es eine Größenbeschränkung von 10 GB pro Partitionsschlüsselwert. Eine Tabelle mit LSIs kann eine beliebige Anzahl von Elementen speichern, solange die Gesamtgröße für einen Partitionsschlüsselwert 10 GB nicht überschreitet.
- Identifizieren Sie die Notwendigkeit [globaler sekundärer Indizes](#) (GSIs). Dies sind Indizes, die einen Partitionsschlüssel und einen Sortierschlüssel haben, die sich von denen in der Basistabelle unterscheiden können (siehe [Blogbeitrag](#) ).
- Definieren Sie die Indexprojektionen. Ziehen Sie die Projizierung einer kleineren Anzahl von Attributen in Betracht, um die Größe der Elemente, die in den Index geschrieben werden, zu minimieren. In diesem Schritt sollten Sie bestimmen, ob Sie Folgendes verwenden möchten:
  - [Sparse-Indizes](#)

- [Materialisierte Aggregationsabfragen](#)
  - [GSI-Überladung](#)
  - [GSI-Sharding](#)
  - [Ein Eventually-Consistent-Replikat mit GSI](#)
- Der Datenbankingenieur bestimmt, ob die Daten große Elemente enthalten. In diesem Fall entwerfen sie die Lösung [mithilfe der Komprimierung oder durch Speichern von Daten](#) in Amazon Simple Storage Service (Amazon S3).
  - Der Datenbankingenieur bestimmt, ob Zeitreihendaten benötigt werden. In diesem Fall verwenden sie das [Zeitreihendesignmuster](#), um die Daten zu modellieren.
  - Der Datenbankingenieur bestimmt, ob das ER-Modell many-to-many Beziehungen enthält. In diesem Fall verwenden sie ein [Adjazenzlisten-Designmuster](#), um die Daten zu modellieren.

## Tools und Ressourcen

- [NoSQL Workbench für Amazon DynamoDB](#) – Bietet Funktionen zur Datenmodellierung, Datenvisualisierung sowie Abfrageentwicklung und -tests, mit denen Sie Ihre DynamoDB-Datenbank entwerfen können
- [NoSQL-Design für DynamoDB](#) (DynamoDB-Dokumentation)
- [Auswahl des richtigen DynamoDB-Partitionsschlüssels](#) (AWS-Datenbank-Blog)
- [Bewährte Methoden für die Verwendung sekundärer Indizes in DynamoDB](#) (DynamoDB-Dokumentation)
- [So entwerfen Sie globale sekundäre Indizes von Amazon DynamoDB](#) (AWS-Datenbank-Blog)

## RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
I	I	I	R/A		

## Outputs

- DynamoDB-Tabellenschema, das Ihre Zugriffsmuster und Anforderungen erfüllt

## Beispiel

Der folgende Screenshot zeigt NoSQL Workbench.

Primary Key		Attributes					
Partition Key: pk	Sort Key: sk						
P1	B1	GS11-PK	GS11-SK	name	desc		
		B1	P1	The Tiki Bundle	Everything you need for an island theme party.		
P4	B2	GS11-PK	GS11-SK	name	desc		
		B2	P4	Tiki Bar Set	Be the Mai Tai master with your very own Tiki Bar.		
P2	B1	name	desc	qty	GS11-PK	GS11-SK	location
		Tiki Torch	Bamboo tiki torch, 4 ft	6	B1	P2	W1-A9-S10-B52
	B2	name	desc	qty	GS11-PK	GS11-SK	location
		Tiki Torch	Bamboo tiki torch, 4 ft	2	B2	P2	W1-A9-S10-B52
	P2	name	desc	qty	location	reorderAt	GS13-SK
		Tiki Torch	Bamboo tiki torch, 4 ft	656	W1-A9-S10-B52	100	/GardenOutdoor/OutdoorDecor/Lighting/LanternsTr
B1	name	desc	qty	GS11-PK	GS11-SK	location	
	Tiki Statue - Pele	Tiki of the Hawaiian Fire Goddess Pele, 5 ft.	1	B1	P3	W1-A15-S6-B27	

## Schritt 6: Erstellen der Datenabfragen

### Ziel

- Erstellen Sie die Hauptabfragen, um das Datenmodell zu validieren.

### Prozess

- Der Datenbankingenieur erstellt manuell eine DynamoDB-Tabelle in der AWS Region oder auf seinem Computer (DynamoDB Local).
- Der Datenbankingenieur fügt der DynamoDB-Tabelle Beispieldaten hinzu.
- Der Datenbankingenieur erstellt Facetten mit dem NoSQL Workbench für Amazon DynamoDB oder dem AWS SDK für Java oder Python, um Beispielabfragen zu erstellen (siehe [Blogbeitrag](#) ).

Facetten sind wie eine Ansicht der DynamoDB-Tabelle.

- Datenbankingenieur und Cloud-Entwickler erstellen Beispielabfragen mithilfe der AWS Command Line Interface (AWS CLI) oder des AWS SDK für die bevorzugte Sprache.

## Tools und Ressourcen

- Ein aktives AWS Konto, um Zugriff auf die DynamoDB-Konsole zu erhalten
- [DynamoDB Local](#) (optional), wenn Sie die Datenbank auf Ihrem Computer erstellen möchten, ohne auf den DynamoDB-Webservice zuzugreifen
- [NoSQL Workbench für Amazon DynamoDB](#) (Download und Dokumentation)
- [AWS SDK](#) in einer Sprache Ihrer Wahl (JavaScript, Python, PHP, .NET, Ruby, Java, Go, Node.js, C++ und SAP ABAP)

## RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
I	I	I	R/A	R	

## Outputs

- Code zum Abfragen der DynamoDB-Tabelle

## Beispiele

- [DynamoDB-Beispiele mit dem AWS SDK for Java](#)
- [Python-Beispiele](#)
- [JavaScript Beispiele](#)

## Schritt 7. Validieren des Datenmodells

### Ziel

- Stellen Sie sicher, dass das Datenmodell Ihren Anforderungen entspricht.

### Prozess

- Der Datenbankingenieur füllt die DynamoDB-Tabelle mit Beispieldaten aus.
- Der Datenbankingenieur führt den Code aus, um die DynamoDB-Tabelle abzufragen.
- Der Datenbankingenieur sammelt die Abfrageergebnisse.
- Der Datenbankingenieur erfasst die Abfrageleistungsmetriken.
- Der Geschäftsbenutzer überprüft, ob die Abfrageergebnisse den Geschäftsanforderungen entsprechen.
- Geschäftsanalysten überprüfen die technischen Anforderungen.

### Tools und Ressourcen

- Ein aktives AWS Konto, um Zugriff auf die DynamoDB-Konsole zu erhalten
- [DynamoDB Local](#) (optional), wenn Sie die Datenbank auf Ihrem Computer erstellen möchten, ohne auf den DynamoDB-Webservice zuzugreifen
- [AWS SDK](#) in einer von Ihnen gewählten Sprache

### RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
A	R	I	C		

### Outputs

- Genehmigtes Datenmodell

## Schritt 8. Überprüfen der Kostenschätzung

### Ziele

- Definieren Sie das Kapazitätsmodell und schätzen Sie die DynamoDB-Kosten, um die Kostenschätzung aus [Schritt 2](#) zu verfeinern.
- Holen Sie sich die endgültige finanzielle Genehmigung vom Geschäftsanalysten und den Stakeholdern.

### Prozess

- Der Datenbankingenieur identifiziert die Schätzung des Datenvolumens.
- Der Datenbankingenieur identifiziert die Datenübertragungsanforderungen.
- Der Datenbankingenieur definiert die erforderlichen Lese- und Schreibkapazitätseinheiten.
- Der Geschäftsanalyst entscheidet zwischen [On-Demand- und bereitgestellten Kapazitätsmodellen](#).
- Der Datenbankingenieur identifiziert die Notwendigkeit von [DynamoDB Auto Scaling](#).
- Der Datenbankingenieur gibt die Parameter im Tool Simple Monthly Calculator ein.
- Der Datenbankingenieur präsentiert den Geschäfts-Stakeholdern die endgültige Preisschätzung.
- Geschäftsanalyst und Stakeholder genehmigen oder lehnen die Lösung ab.

### Tools und Ressourcen

- [AWS Pricing Calculator](#)

### RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
C	A	I	R		

## Outputs

- Kapazitätsmodell
- Überarbeitete Kostenschätzung

## Schritt 9. Bereitstellen des Datenmodells

### Ziel

- Stellen Sie die DynamoDB-Tabelle (oder Tabellen) in der bereitAWS-Region.

### Prozess

- DevOps Der -Architekt erstellt eine -AWS CloudFormationVorlage oder ein anderes Infrastructure as Code (IaC)-Tool für die DynamoDB-Tabelle (oder -Tabellen). AWS CloudFormation bietet eine automatisierte Möglichkeit, Ihre Tabellen und zugehörigen Ressourcen bereitzustellen und zu konfigurieren.IaC

### Tools und Ressourcen

- [AWS CloudFormation](#)

### RACI

Geschäftsbenutzer	Geschäftsanalyst	Lösungsarchitekt	Datenbankingenieur	Anwendungsentwickler	DevOps Techniker
I	I	C	C		R/A

## Outputs

- AWS CloudFormation-Vorlage

## Beispiel

```
mySecondDDBTable:
  Type: AWS::DynamoDB::
  Table DependsOn: "myFirstDDBTable"
  Properties:
    AttributeDefinitions:
      - AttributeName: "ArtistId"
        AttributeType: "S"
      - AttributeName: "Concert"
        AttributeType: "S"
      - AttributeName: "TicketSales"
        AttributeType: "S"
    KeySchema:
      - AttributeName: "ArtistId"
        KeyType: "HASH"
      - AttributeName: "Concert"
        KeyType: "RANGE"
    ProvisionedThroughput:
      ReadCapacityUnits:
        Ref: "ReadCapacityUnits"
      WriteCapacityUnits:
        Ref: "WriteCapacityUnits"
    GlobalSecondaryIndexes:
      - IndexName: "myGSI"
        KeySchema:
          - AttributeName: "TicketSales"
            KeyType: "HASH"
        Projection:
          ProjectionType: "KEYS_ONLY"
        ProvisionedThroughput:
          ReadCapacityUnits:
            Ref: "ReadCapacityUnits"
          WriteCapacityUnits:
            Ref: "WriteCapacityUnits"
    Tags:
      - Key: mykey
        Value: myvalue
```

# Vorlagen

Die in diesem Abschnitt bereitgestellten Vorlagen basieren auf den [Modellierung von Game Player-Daten mit Amazon DynamoDB](#) auf der -AWSWebsite.

## Note

Die Tabellen in diesem Abschnitt verwenden MM als Abkürzung für Millionen und K als Abkürzung für Tausende.

## Themen

- [Bewertungsvorlage für Geschäftsanforderungen](#)
- [Bewertungsvorlage für technische Anforderungen](#)
- [Vorlage für Zugriffsmuster](#)

## Bewertungsvorlage für Geschäftsanforderungen

Geben Sie eine Beschreibung für den Anwendungsfall ein:

## Beschreibung

Angenommen, Sie erstellen ein Online-Multiplayer-Spiel. In Ihrem Spiel treten Gruppen von 50 Spielern einer Sitzung bei, um ein Spiel zu spielen, was normalerweise etwa 30 Minuten dauert. Während des Spiels müssen Sie den Datensatz eines bestimmten Spielers aktualisieren, um die Zeit anzugeben, die der Spieler gespielt hat, seine Statistiken oder ob er das Spiel gewinnen wird. Benutzer möchten frühere Spiele sehen, die sie gespielt haben, entweder um die Stimmungen der Spiele anzuzeigen oder um eine Wiederholung der Aktion jedes Spiels zu sehen.

Geben Sie Informationen zu Ihren Benutzern an:

Nutzer

Beschreibung

Erwartete Zahl

Game Player	Online-Spieler.	1 MM
Entwicklungsteam	Internes Team, das die Spielstatistiken zur Verbesserung der Spielerfahrung verwendet.	100

Geben Sie Informationen zu den Datenquellen an und wie Daten aufgenommen werden:

Quelle	Beschreibung	Nutzer
Online-Spiel	Spieler erstellen Profile und starten neue Spiele.	Game Player
Spiel-App	Die Spiel-App erfasst automatisch Statistiken über die Spiele, z. B. Start- und Endzeit, Anzahl der Spieler, Position jedes Spielers und Karte für das Spiel.	

Geben Sie Informationen darüber an, wie Daten verbraucht werden:

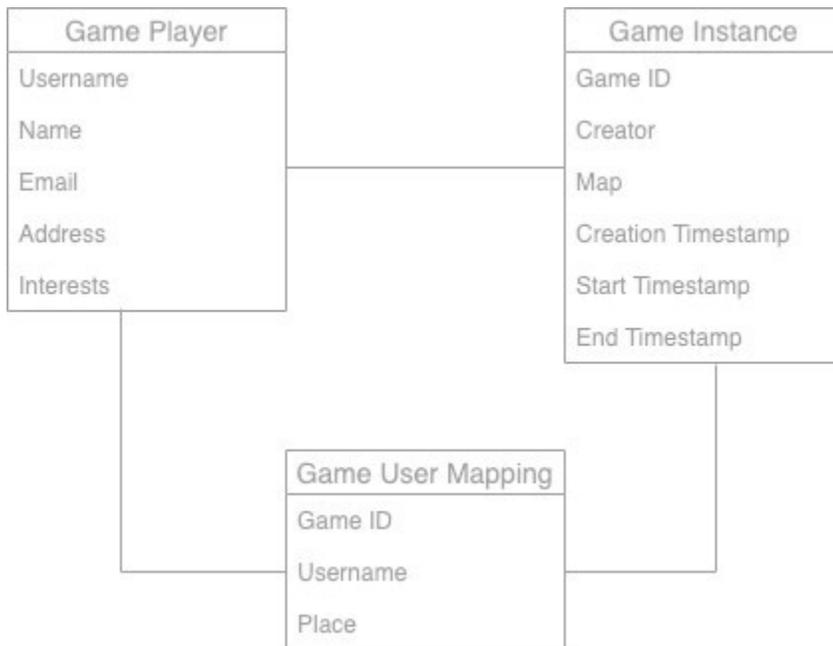
Verbraucher	Beschreibung	Nutzer
Online-Spiel	Spieler sehen sich Profile an und überprüfen ihre Spielstatistiken.	Game Player
Datenanalysen	Das Spielentwicklungsteam extrahiert Spielstatistiken für Datenanalysen und zur Verbesserung der Benutzererfahrung. Daten werden aus dem Datenspeicher exportiert.	Entwicklungsteam

und in Amazon S3 importiert,  
um Analysen über eine Spark-  
Anwendung zu unterstützen.

Geben Sie eine Liste der Entitäten an und wie sie identifiziert werden:

Entitätsname	Beschreibung	ID
Game Player	Speichert Informationen wie Identifizierung, Adresse, demografische Daten, Interessen für jeden Benutzer (Gamer).	Username
Spiel-Instance	Stellt Informationen zu jedem gespielten Spiel bereit, einschließlich Ersteller, Start, Ende und Karten-Yplayed.	Spiel-ID
Zuordnung von Spielbenutzern	Stellt die many-to-many Beziehungen zwischen Benutzern und Spielen dar.	Spiel-ID UND Benutzername

Erstellen Sie ein ER-Modell für die Entitäten:



Stellen Sie allgemeine Statistiken zu den Entitäten bereit:

Entity Name	Geschätzte Anzahl von Datensätzen	Datensatzgröße	Hinweise
Game Player	1 MM	< 1 KB	Die Spielplattform hat etwa 1 MM-Benutzer.
Spiel-Instance	6 MM (100.000 K/Tag * 60 Tage)	< 1 KB	Im Durchschnitt gibt es täglich 100K Spiele. Wir müssen die letzten 60 Tage speichern.
Zuordnung von Spielbenutzern	300 MM (6 MM-Spiele * 50 Spieler)	< 1 KB	Im Durchschnitt hat jedes Spiel 50 Spieler, zu denen wir Informationen speichern müssen.

# Bewertungsvorlage für technische Anforderungen

Geben Sie Informationen zu Datenerfassungstypen an:

Datenaufnahmetyp	J/N	Beschreibung	Frequency (Frequenz)
Anwendungszugriff	Y		
API-Gateway	Y		
Daten-Streaming	N		
Batch-Prozess	N		
ETL	N		
Datenimport	N		
Zeitreihen	N		

Geben Sie Informationen zu Datennutzungstypen an:

Datennutzungstyp	J/N	Beschreibung	Frequency (Frequenz)
Anwendungszugriff			
API-Gateway			
Datenexport			
Datenanalysen			
Datenaggregation			
Berichterstellung			
Suche			
Daten-Streaming			

## ETL

Geben Sie Schätzungen des Datenvolumens an:

Entitätsname	Geschätzte Anzahl von Datensätzen	Datensatzgröße	Datenvolumen
Game Player	1 MM	< 1 KB	~ 1 GB (1 MM * 1 KB)
Spiel-Instance	6 MM (100 000/Tag * 60 Tage)	< 1 KB	~ 6 GB (6 MM * 1 KB)
Zuordnung von Spielbenutzern	300 MM (6 MM-Spiele * 50 Spieler)	< 1 KB	~ 300 GB (300 MM * 1 KB)

 Note

Der Aufbewahrungszeitraum für Daten beträgt 60 Tage. Nach 60 Tagen müssen Daten für Analysen in Amazon S3 gespeichert werden, indem [DynamoDB Time to Live \(TTL\)](#) verwendet wird, um Daten automatisch aus DynamoDB nach Amazon S3 zu verschieben.

Beantworten Sie diese Fragen zu Zeitmustern:

- In welchem Zeitraum steht die Anwendung dem Benutzer zur Verfügung (z. B. 24.00 Uhr oder 9.00 Uhr bis 17.00 Uhr an Wochentagen)?
- Gibt es am Tag einen Spitzenwert? Wie viele Stunden? Wie hoch ist der Prozentsatz der Anwendungsnutzung?

Geben Sie die Anforderungen an den Schreibdurchsatz an:

Entitätsname	Schreibvorgänge/Tag	Stunden/Tag	Schreibvorgänge/Sekunde
Game Player	10 000 Updates	18	< 1
Spiel-Instance	300 000	18	< 5
Zuordnung von Spielbenutzern	1 800 000 000	18	~ 27,777

### Hinweise

Game-Player-Schreibvorgänge: 1 Prozent der Benutzer aktualisieren ihre Profile täglich, daher erwarten wir 10 000 Updates für 1 000 000 Benutzer.

Schreibvorgänge für Game Instances: 100.000 Spiele/Tag. Für jedes Spiel haben wir mindestens 3 Schreibvorgänge – bei der Erstellung, beim Start und am Ende –, sodass die Gesamtzahl 300.000 Schreibvorgänge beträgt.

Schreibvorgänge für Game User Mapping: 100.000 Spiele/Tag für jedes Spiel mit 50 Spielern. Die durchschnittliche Spieldauer beträgt 30 Minuten, und die Spielposition wird alle 5 Sekunden aktualisiert. Wir schätzen einen Durchschnitt von 360 Updates pro Spieler, sodass die Summe  $100.000 * 50 * 360 = 1.800.000.000$  Schreibvorgänge beträgt.

Geben Sie die Anforderungen an den Lesedurchsatz an:

Entitätsname	Lesevorgänge/Tag	Stunden/Tag	Lesevorgänge/Sek.
Game Player	200 000	18	~ 3
Spiel-Instance	5 000 000	18	~ 77
Zuordnung von Spielbenutzern	1 800 000 000	18	~ 27,777

### Hinweise

Game Player-Lesevorgänge: 20 Prozent der Benutzer starten Spiele, also  $1 \text{ MM} * 0,2 = 200.000$ .

Lesevorgänge für Game Instances: 100.000 Spiele/Tag. Für jedes Spiel haben wir mindestens eine Leseoperation pro Spieler und 50 Spieler pro Spiel, sodass die Gesamtzahl 5 000 000 Leseoperationen beträgt.

Lesevorgänge für Game User Mapping: 100.000 Spiele/Tag für 50 Spieler. Die durchschnittliche Spieldauer beträgt 30 Minuten, und die Spielposition wird alle 5 Sekunden aktualisiert. Wir schätzen einen Durchschnitt von 360 Updates pro Spieler, und jedes Update erfordert einen Lesevorgang, sodass der Gesamtwert  $100.000 * 50 * 360 = 1.800.000.000$  Lesevorgänge beträgt.

Geben Sie die Latenzanforderungen für den Datenzugriff an:

Operation	99 Perzentile	Maximale Latenz
Read (Lesen)	30 ms	100 ms
Write (Schreiben)	10 ms	50 ms

Geben Sie die Anforderungen an die Datenverfügbarkeit an:

Anforderung	J/N	Metrik	Hinweise
Hohe Verfügbarkeit	Y	99,9 %	
RTO	Y	1 Stunde	Recovery Time Objective
RPO	Y	1 Stunde	Recovery Point Objective
Notfallwiederherstellung	N		

In-Region-Datenreplikation	N		
Regionsübergreifende Datenreplikation	N	Latenz von 3 Sekunden	Welche AWS-Regionen?

Geben Sie die Sicherheitsanforderungen an:

Anforderung	J/N	Hinweise
Sensibler Datenspeicher	N	Geschützte Gesundheitsdaten (PHI), Informationen zur Zahlungskartenbranche (PCI), persönlich identifizierbare Informationen (PII)?
Verschlüsselung im Ruhezustand	Y	
Verschlüsselung während der Übertragung	Y	
Clientseitige Verschlüsselung	N	
Jede proprietäre Bibliothek oder Verschlüsselungsbibliothek eines Drittanbieters	N	
Protokollierung des Datenzugriffs	N	
Prüfung des Datenzugriffs	N	

## Vorlage für Zugriffsmuster

Sammeln und dokumentieren Sie Informationen zu den Zugriffsmustern für den Anwendungsfall mithilfe der folgenden Felder:

Feld	Beschreibung
Zugriffsmuster	Geben Sie einen Namen für das Zugriffsmuster an.
Beschreibung	Geben Sie eine detailliertere Beschreibung des Zugriffsmusters an.
Priorität	Definieren Sie eine Priorität für das Zugriffsmuster (hoch, mittel oder niedrig). Dadurch werden die relevantesten Zugriffsmuster für die Anwendung definiert.
Lesen oder Schreiben	Ist es ein Lesezugriff oder Schreibzugriffsmuster?
Typ	greift das Muster auf ein einzelnes Element, mehrere Elemente oder alle Elemente zu?
Filter	Benötigt das Zugriffsmuster einen Filter?
Sortierung	Erfordert das Ergebnis eine Sortierung?

## Vorlage

Zugriffsmuster	Beschreibung	Priorität	Lesen oder Schreiben	Typ (Einzelement, mehrere Elemente oder alle)	Schlüsselattribut	Filter	Ergebnisreihenfolge
Benutzerprofil erstellen	Der Benutzer erstellt	Hoch	Write (Schreiben)	Einzelner Artikel	Username	N/A	N/A

	ein neues Profil.						
Benutzerprofil aktualisieren	Der Benutzer aktualisiert sein Profil.	Mittel	Write (Schreiben)	Einzelner Artikel	Username	Benutzername = aktueller Benutzer	N/A
Benutzerprofil abrufen	Der Benutzer überprüft sein Profil.	Hoch	Read (Lesen)	Einzelner Artikel	Username	Benutzername = aktueller Benutzer	N/A
Erstellen eines Spiels	Der Benutzer erstellt ein neues Spiel.	Hoch	Write (Schreiben)	Einzelner Artikel	GameID	N/A	N/A

Suchen nach offenen Spielen	Der Benutzer sucht nach offenen Spielen. Die Suchergeb nisse werden nach Startzeit stempel in absteigen der Reihenfol ge sortiert.	Hoch	Read (Lesen)	Mehrere Elemente	GameStatu s = geöffnet	Startzeit stempel- A bstammung
--------------------------------------	--	------	-----------------	---------------------	------------------------------	---

Suchen nach offenen Spielen nach Karte	Der Benutzer sucht nach offenen Spielen, indem er eine bestimmte Zuordnung verwendet, die nach Startzeit stempel in absteigen der Reihenfolge sortiert ist Reihenfolge.	Mittel	Read (Lesen)	Mehrere Elemente		GameStatus = geöffnet und Map = XYZ	Startzeitstempel-Abstammung
Spiel anzeigen	Der Benutzer überprüft die Details eines Spiels.	Hoch	Read (Lesen)	Einzelner Artikel	GameID	N/A	N/A

Benutzer in einem Spiel anzeigen	Der Benutzer erhält eine Liste aller Benutzer in einem Spiel.	Mittel	Read (Lesen)	Mehrere Elemente	GameID = XYZ	N/A	
Benutzer zu einem Spiel hinzufügen	Der Benutzer tritt einem offenen Spiel bei.	Hoch	Write (Schreiben)	Einzelner Artikel	GameID und Benutzername	GameStatus = geöffnet	N/A
Starten eines Spiels	Der Benutzer startet ein neues Spiel.	Hoch	Write (Schreiben)	Einzelner Artikel	GameID	N/A	N/A
Aktualisieren des Spiels für den Benutzer	Aktualisieren Sie die Benutzerposition im Spiel.	Mittel	Write (Schreiben)	Einzelner Artikel	GameID und Benutzername	N/A	N/A
Aktualisieren des Spiels	Spiel endet; Statistiken aktualisieren.	Mittel	Write (Schreiben)	Einzelner Artikel	GameID	N/A	N/A

Alle vergangen en Spiele für einen Benutzer finden	Listen Sie alle Spiele auf, die ein Benutzer nach dem Startzeit stempel des Spiels geordnet gespielt hat.	Niedrig	Read (Lesen)	Mehrere Elemente	Benutzern ame und GameID	Benutzern ame = aktueller Benutzer	Startzeit stempel
Exportier en von Daten für Datenanal ysen	Das Entwicklu ngsteam führt einen Batch- Job durch, um Daten nach Amazon S3 zu exportier en.	Niedrig	Read (Lesen)	Alle	N/A	N/A	N/A

# Bewährte Methoden

Erwägen Sie die Verwendung der folgenden bewährten Methoden für das DynamoDB-Design:

- [Entwerfen von Partitionsschlüsseln](#) – Verwenden Sie einen Partitionsschlüssel mit hoher Kardinalität, um die Last gleichmäßig zu verteilen.
- [Adjazenzlisten-Designmuster](#) – Verwenden Sie dieses Designmuster für die Verwaltung von one-to-many und many-to-many Beziehungen.
- [Sparse-Index](#) – Verwenden Sie den Sparse-Index für Ihre globalen sekundären Indizes (GSIs). Wenn Sie einen GSI erstellen, geben Sie einen Partitionsschlüssel und optional einen Sortierschlüssel an. Nur Elemente in der Basistabelle, die einen entsprechenden GSI-Partitionsschlüssel enthalten, werden im Sparse-Index angezeigt. Dies trägt dazu bei, die GSIs kleiner zu halten.
- [Überladen von Indizes](#) – Verwenden Sie dieselbe GSI für die Indizierung verschiedener Arten von Elementen.
- [GSI Write-Sharding](#)— Sharden Sie mit Bedacht, um Daten auf die Partitionen zu verteilen und so effiziente und schnellere Abfragen zu ermöglichen.
- [Große Gegenstände](#) – Speichern Sie nur Metadaten in der Tabelle, speichern Sie den Blob in Amazon S3 und behalten Sie die Referenz in DynamoDB bei. Teilen Sie große Elemente in mehrere Elemente auf und indizieren Sie sie effizient mithilfe von Sortierschlüsseln.

Weitere bewährte Methoden für das Design finden Sie in der [Amazon-DynamoDB-Dokumentation](#).

# Beispiel für hierarchische Datenmodellierung

In den folgenden Abschnitten wird ein Beispiel für ein Automobilunternehmen verwendet, um zu zeigen, wie Sie die Schritte des Datenmodellierungsprozesses verwenden können, um ein mehrstufiges Komponentenverwaltungssystem in DynamoDB zu entwerfen.

## Themen

- [Schritt 1: Identifizieren der Anwendungsfälle und des logischen Datenmodells](#)
- [Schritt 2: Erstellen einer vorläufigen Kostenschätzung](#)
- [Schritt 3: Identifizieren Ihrer Datenzugriffsmuster](#)
- [Schritt 4: Identifizieren der technischen Anforderungen](#)
- [Schritt 5: Erstellen eines DynamoDB-Datenmodells](#)
- [Schritt 6: Erstellen von Datenabfragen](#)
- [Schritt 7: Validieren des Datenmodells](#)
- [Schritt 8: Überprüfen der Kostenschätzung](#)
- [Schritt 9: Bereitstellen des Datenmodells](#)

## Schritt 1: Identifizieren der Anwendungsfälle und des logischen Datenmodells

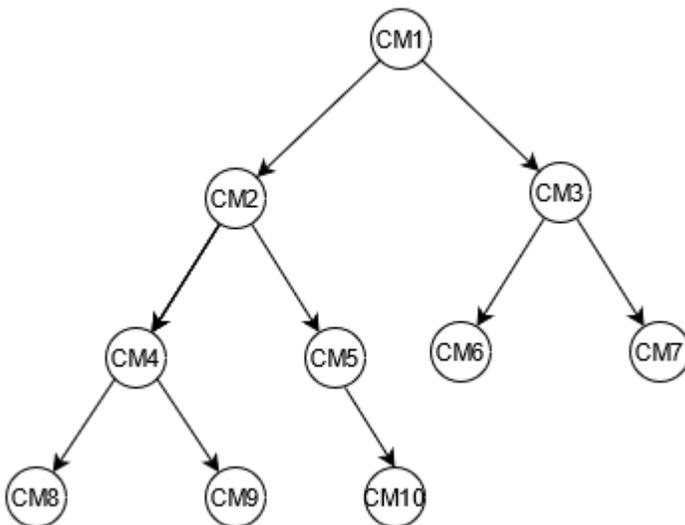
Ein Automobilunternehmen möchte ein Verwaltungssystem für Transaktionskomponenten aufbauen, um alle verfügbaren Autoteile zu speichern und nach ihnen zu suchen und Beziehungen zwischen verschiedenen Komponenten und Teilen aufzubauen. Ein Auto enthält beispielsweise mehrere Batterien, jede Batterie enthält mehrere High-Level-Module, jedes Modul enthält mehrere Zellen und jede Zelle enthält mehrere Low-Level-Komponenten.

Im Allgemeinen ist für den Aufbau eines hierarchischen Beziehungsmodells eine Graphdatenbank wie [Amazon Neptune](#) eine bessere Wahl. In einigen Fällen ist Amazon DynamoDB jedoch aufgrund seiner Flexibilität, Sicherheit, Leistung und Skalierbarkeit eine bessere Alternative für die hierarchische Datenmodellierung.

Sie könnten beispielsweise ein System erstellen, in dem 80–90 Prozent der Abfragen transaktionaler Natur sind, wofür DynamoDB gut geeignet ist. In diesem Beispiel sind die anderen 10–20 Prozent

der Abfragen relational, wo eine Graphdatenbank wie Neptune besser passt. In diesem Fall könnte die Aufnahme einer zusätzlichen Datenbank in die Architektur, um nur 10–20 Prozent der Abfragen zu erfüllen, die Kosten erhöhen. Es erhöht auch den betrieblichen Aufwand, der mit der Wartung mehrerer Systeme und der Synchronisierung der Daten verbunden ist. Stattdessen können Sie diese 10 bis 20 Prozent relationalen Abfragen in DynamoDB modellieren.

Wenn Sie einen Beispielbaum für Fahrzeugkomponenten grafisch darstellen, können Sie die Beziehung zwischen diesen Komponenten besser abbilden. Das folgende Diagramm zeigt eine Abhängigkeitsgrafik mit vier Stufen. CM1 ist die oberste Komponente für das Beispielfahrzeug selbst. Sie besteht aus zwei Unterkomponenten für zwei Beispielbatterien, CM2 und CM3. Jede Batterie hat zwei Unterkomponenten, nämlich die Module. CM2 hat die Module CM4 und CM5 und CM3 hat die Module CM6 und CM7. Jedes Modul hat mehrere Unterkomponenten, nämlich die Zellen. Das CM4-Modul hat zwei Zellen, CM8 und CM9. CM5 hat eine Zelle, CM10. CM6 und CM7 haben noch keine zugehörigen Zellen.



In diesem Handbuch werden dieser Baum und seine Komponenten-IDs als Referenz verwendet. Eine der obersten Komponenten wird als Elternteil bezeichnet und eine Unterkomponente wird als Kind bezeichnet. Beispielsweise ist die oberste Komponente CM1 das Elternteil von CM2 und CM3. CM2 ist das Elternteil von CM4 und CM5. Dadurch werden die Eltern-Kind-Beziehungen grafisch dargestellt.

In der Baumstruktur können Sie das vollständige Abhängigkeitsdiagramm einer Komponente sehen. CM8 ist beispielsweise von CM4 abhängig, das von CM2 abhängig ist, das von CM1 abhängig ist. Der Baum definiert den vollständigen Abhängigkeitsgraphen als Pfad. Ein Pfad beschreibt zwei Dinge:

- Das Abhängigkeitsdiagramm
- Die Position im Baum

Ausfüllen der Vorlagen für Geschäftsanforderungen:

Geben Sie Informationen zu Ihren Benutzern an:

Nutzer	Beschreibung
Mitarbeiter	Interner Mitarbeiter des Automobilunternehmens, der Informationen über Autos und seine Komponenten benötigt

Geben Sie Informationen zu den Datenquellen an und wie Daten aufgenommen werden:

Quelle	Beschreibung	Nutzer
Verwaltungssystem	System, das alle Daten im Zusammenhang mit verfügbaren Autoteilen und deren Beziehungen zu anderen Komponenten und Teilen speichert.	Mitarbeiter

Geben Sie Informationen darüber an, wie Daten verbraucht werden:

Verbraucher	Beschreibung	Nutzer
Verwaltungssystem	Rufen Sie alle unmittelbar untergeordneten Komponenten für eine übergeordnete Komponenten-ID ab.	Mitarbeiter
Verwaltungssystem	Rufen Sie eine rekursive Liste aller untergeordneten	Mitarbeiter

Komponenten für eine  
Komponenten-ID ab.

Verwaltungssystem

Sehen Sie sich die Vorgänger Mitarbeiter  
einer Komponente an.

## Schritt 2: Erstellen einer vorläufigen Kostenschätzung

Es ist wichtig, eine Schätzung der Kosten für alle Umgebungen Ihrer Anwendung zu berechnen, damit Sie überprüfen können, ob die Lösung wirtschaftlich tragbar ist. Eine bewährte Methode besteht darin, eine allgemeine Schätzung vorzunehmen und die Genehmigung des Geschäftsanalysten zu erhalten, bevor Sie mit der Entwicklung und Bereitstellung fortfahren.

- Der Datenbankingenieur erstellt die erste Kostenanalyse anhand verfügbarer Informationen und der Beispiele, die auf der [DynamoDB-Preisseite](#) dargestellt werden.
  - Erstellen Sie eine Kostenschätzung für On-Demand-Kapazität (siehe [Beispiel](#)).
  - Erstellen Sie eine Kostenschätzung für bereitgestellte Kapazität (siehe [Beispiel](#)).
    - Rufen Sie für das bereitgestellte Kapazitätsmodell die geschätzten Kosten vom Rechner ab und wenden Sie den Rabatt für reservierte Kapazität an.
  - Vergleichen Sie die geschätzten Kosten der beiden Kapazitätsmodelle.
  - Erstellen Sie eine Schätzung für alle Umgebungen (Dev, Prod, QA).
- Geschäftsanalyst überprüft und genehmigt oder lehnt die vorläufige Kostenschätzung ab.

Anhand dieser Referenzwerte können Sie einen geschätzten Preis erstellen, der zur Genehmigung eingereicht werden soll. Um das Budget zu erstellen, können Sie die [DynamoDB-Preisseite](#) und [AWS-Preisrechner](#) verwenden.

## Schritt 3: Identifizieren Ihrer Datenzugriffsmuster

Dieser Beispielanwendungsfall hat die folgenden Zugriffsmuster für die Verwaltung von Beziehungen zwischen verschiedenen Fahrzeugkomponenten.

Zugriffsmuster	Priorität	Lesen oder Schreiben	Beschreibung	Typ	Filter	Ergebnisreihenfolge

Sofortiges untergeordnetes Element	Hoch	Lesen	Rufen Sie alle unmittelbar untergeordneten Komponenten für eine übergeordnete Komponenten-ID ab.	Mehrere	Component ID	N/A
Alle untergeordneten Komponenten	Hoch	Lesen	Rufen Sie eine rekursive Liste aller untergeordneten Komponenten für eine Komponenten-ID ab.	Mehrere	Component ID	N/A
Vorgänger	Hoch	Lesen	Rufen Sie die Vorgänger einer Komponente ab.	Mehrere	Component ID	N/A

## Schritt 4: Identifizieren der technischen Anforderungen

Dieses Beispiel hat keine spezifischen technischen Anforderungen, die außerhalb des Geltungsbereichs dieses Beispiels liegen. In realen Fällen ist es eine bewährte Methode, diesen Schritt abzuschließen und zu überprüfen, ob alle technischen Anforderungen erfüllt sind, bevor Sie mit der Entwicklung und Bereitstellung fortfahren. Sie können diesen Schritt in Ihrem

Geschäftsfall mit dem [Beispiel-Prototyp](#) abschließen. Darüber hinaus empfehlen wir, die [DynamoDB-Servicekontingente](#) zu validieren, um sicherzustellen, dass Ihre konzipierte Lösung keine harten Limits enthält.

## Schritt 5: Erstellen eines DynamoDB-Datenmodells

Definieren Sie die Partitionsschlüssel für Ihre Basistabelle und globale sekundäre Indizes (GSIs):

- Verwenden Sie gemäß den wichtigsten bewährten Designmethoden `ComponentId` als Partitionsschlüssel für die Basistabelle in diesem Beispiel. Da es einzigartig ist, `ComponentId` kann Granularität bieten. DynamoDB verwendet den Hashwert Ihres Partitionsschlüssels, um die Partition zu ermitteln, auf der die Daten physisch gespeichert sind. Die eindeutige Komponenten-ID generiert einen anderen Hashwert, der die Verteilung der Daten innerhalb der Tabelle erleichtern kann. Sie können die Basistabelle mithilfe eines `ComponentId` Partitionsschlüssels abfragen.
- Um unmittelbare untergeordnete Elemente einer Komponente zu finden, erstellen Sie einen GSI, wobei `ParentId` der Partitionsschlüssel und der Sortierschlüssel `ComponentId` ist. Sie können diesen GSI abfragen, indem Sie `ParentId` als Partitionsschlüssel verwenden.
- Um alle rekursiven einer Komponente zu finden, erstellen Sie eine GSI, wobei `GraphId` der Partitionsschlüssel ist und `Path` der Sortierschlüssel ist. Sie können diesen GSI abfragen, indem Sie `GraphId` als Partitionsschlüssel und den `BEGINS_WITH(Path, "$path")`-Operator auf dem Sortierschlüssel verwenden.

	Partitionsschlüssel	Sortierschlüssel	Zuordnungsattribute
Basistabelle	<code>ComponentId</code>		<code>ParentId</code> , <code>GraphId</code> , <code>Path</code>
GSI1	<code>ParentId</code>	<code>ComponentId</code>	
GSI2	<code>GraphId</code>	<code>Path</code>	<code>ComponentId</code>

## Speichern von Komponenten in der Tabelle

Der nächste Schritt besteht darin, jede Komponente in der DynamoDB-Basistabelle zu speichern. Nachdem Sie alle Komponenten aus dem Beispielbaum eingefügt haben, erhalten Sie die folgende Basistabelle.

ComponentId	ParentId	GraphId	Pfad
CM1		CM1#1	CM1
CM2	CM1	CM1#1	CM1 CM2
CM3	CM1	CM1#1	CM1 CM3
CM4	CM2	CM1#1	CM1 CM2 CM4
CM5	CM2	CM1#1	CM1 CM2 CM5
CM6	CM3	CM1#1	CM1 CM3 CM6
CM7	CM3	CM1#1	CM1 CM3 CM7
CM8	CM4	CM1#1	CM1 CM2 CM4 CM8
CM9	CM4	CM1#1	CM1 CM2 CM4 CM9
CM10	CM5	CM1#1	CM1 CM2 CM5 CM10

## Der GSI1-Index

Um alle unmittelbar untergeordneten Elemente einer Komponente zu überprüfen, erstellen Sie einen Index, der `ParentId` als Partitionsschlüssel und `ComponentId` als Sortierschlüssel verwendet. Die folgende Pivot-Tabelle stellt den GSI1-Index dar. Sie können diesen Index verwenden, um alle unmittelbar untergeordneten Komponenten mithilfe einer übergeordneten Komponenten-ID abzurufen. Sie können beispielsweise herausfinden, wie viele Batterien in einem Auto verfügbar sind (CM1) oder welche Zellen in einem Modul (CM4) verfügbar sind.

ParentId	ComponentId
CM1	CM2
	CM3
	CM4
CM2	CM5
	CM6
CM3	CM7
	CM8
CM4	CM9
	CM10
CM5	

## Der GSI2-Index

Die folgende Pivot-Tabelle stellt den GSI2-Index dar. Sie ist mit `GraphId` als Partitionsschlüssel und `Path` als Sortierschlüssel konfiguriert. Mithilfe von `GraphId` und der `-begins_with`Operation für den Sortierschlüssel (`Path`) können Sie die vollständige Herkunft einer Komponente in einem Baum finden.

GraphId	Pfad	ComponentId
CM1#1	CM1	CM1

CM1 CM2	CM2
CM1 CM3	CM3
CM1 CM2 CM4	CM4
CM1 CM2 CM5	CM5
CM1 CM2 CM4 CM8	CM8
CM1 CM2 CM4 CM9	CM9
CM1 CM2 CM5 CM10	CM10
CM1 CM3 CM6	CM6
CM1 CM3 CM7	CM7

## Schritt 6: Erstellen von Datenabfragen

Nachdem Sie Ihre Zugriffsmuster definiert und Ihr Datenmodell entwerfen haben, können Sie hierarchische Daten in der DynamoDB-Datenbank abfragen. Als bewährte Methode, Kosten zu sparen und die Leistung sicherzustellen, verwenden die folgenden Beispiele nur den Abfragevorgang ohne Scan.

- Suchen Sie nach Vorgängern einer Komponente.

Um die Vorfahren (Eltern, Großeltern, Urgroßeltern usw.) der CM8-Komponente zu finden, fragen Sie die Basistabelle mit `ComponentId = "CM8"` ab. Die Abfrage gibt die folgenden Ergebnisse zurück.

Um die Größe der Ergebnisdaten zu reduzieren, können Sie einen Projektionsausdruck verwenden, um nur das `Path`-Attribut zurückzugeben.

ComponentId	ParentId	GraphId	Path
CM8	CM4	CM1#1	CM1 CM2 CM4 CM8

Pfad

CM1|CM2|CM4|CM8

Teilen Sie nun den Pfad mithilfe der Pipe („|“) auf und nehmen Sie die ersten N-1-Komponenten, um Vorfahren zu erhalten.

Abfrageergebnis: Die Vorfahren von CM8 sind CM1, CM2, CM4.

- Finden Sie unmittelbare untergeordnete Elemente einer Komponente.

Um alle unmittelbar untergeordneten oder einstufigen Downstream-Komponenten für die CM2-Komponente abzurufen, fragen Sie GSI1 mit `abParentId = "CM2"`. Die Abfrage gibt die folgenden Ergebnisse zurück.

ParentId	ComponentId
CM2	CM4
	CM5

- Suchen Sie alle untergeordneten Downstream-Komponenten mithilfe einer Komponente der obersten Ebene.

Um alle untergeordneten oder nachgelagerten Komponenten für die Komponente CM1 der obersten Ebene abzurufen, fragen Sie GSI2 mit `GraphId = "CM1#1"` und `abbegins_with("Path", "CM1|")` und verwenden Sie einen Projektionsausdruck mit `ComponentId`. Es werden alle Komponenten zurückgegeben, die sich auf diesen Baum beziehen.

Dieses Beispiel hat einen einzelnen Baum mit CM1 als oberster Komponente. In Wirklichkeit könnten Sie Millionen von Komponenten der obersten Ebene in derselben Tabelle haben.

GraphId	ComponentId
	CM2
CM1#1	CM3

CM4

CM5

CM8

CM9

CM10

CM6

CM7

- Suchen Sie alle untergeordneten Downstream-Komponenten mithilfe einer Komponente der mittleren Ebene.

Um alle untergeordneten oder nachgelagerten Komponenten rekursiv für Komponente CM2 abzurufen, haben Sie zwei Möglichkeiten. Sie können rekursiv Ebene für Ebene abfragen, oder Sie können den GSI2-Index abfragen.

- Fragen Sie GSI1 Ebene für Ebene rekursiv ab, bis die letzte Ebene der untergeordneten Komponenten erreicht ist.

1. Fragen Sie GSI1 mit `ParentId = "CM2"` ab. Das wird die folgenden Ergebnisse zurückgeben.

ParentId	ComponentId
CM2	CM4
	CM5

2. Fragen Sie GSI1 erneut mit `ParentId = "CM4"` ab. Das wird die folgenden Ergebnisse zurückgeben.

ParentId	ComponentId
CM4	CM8
	CM9

3. Fragen Sie GSI1 erneut mit `ParentId = "CM5"` ab. Das wird die folgenden Ergebnisse zurückgeben.

Setzen Sie die Schleife fort: Fragen Sie nach jedem `ComponentId` ab bis Sie die letzte Ebene erreicht haben. Wenn eine Abfrage mit `ParentId = "<ComponentId>"` keine Ergebnisse zurückgibt, stammt das vorherige Ergebnis aus der letzten Ebene des Baums.

ParentId	ComponentId
CM5	CM10

4. Führen Sie alle Ergebnisse zusammen.

```
result=[CM4, CM5] + [CM8, CM9] + [CM10]
      = [CM4, CM5, CM8, CM9, CM10]
```

- Fragen Sie GSI2 ab, in der ein hierarchischer Baum für eine Komponente der obersten Ebene (ein Auto oder CM1) gespeichert wird.
  1. Suchen Sie zunächst die Komponente der obersten Ebene oder den obersten Vorfahren und Path von CM2. Fragen Sie dazu die Basistabelle ab, indem Sie `ComponentId = "CM2"` verwenden, um den Pfad dieser Komponente in der hierarchischen Struktur zu finden. Wählen Sie die Attribute `GraphId` und `Path` aus. Die Abfrage gibt die folgenden Ergebnisse zurück.

GraphId	Pfad
CM1#1	CM1 CM2

2. Fragen Sie GSI2 mit `abGraphId = "CM1#1" AND BEGINS_WITH("Path", "CM1|CM2|")`. Die Abfrage wird die folgenden Ergebnisse zurückgeben.

GraphId	Pfad	ComponentId
---------	------	-------------

CM1#1	CM1 CM2 CM4	CM4
	CM1 CM2 CM5	CM5
	CM1 CM2 CM4 CM8	CM8
	CM1 CM2 CM4 CM9	CM9
	CM1 CM2 CM5 CM10	CM10

3. Wählen Sie das ComponentId-Attribut, um alle untergeordneten Komponenten für CM2 zurückzugeben.

## Schritt 7: Validieren des Datenmodells

In diesem Schritt validiert der Geschäftsbenuer die Abfrageergebnisse und prüft, ob er die Geschäftsanforderungen erfüllt. Sie können die folgende Tabelle verwenden, um die Zugriffsmuster anhand der Anforderungen des Benutzers zu überprüfen.

Frage	Basistabelle/GSI	Abfrage
Als Benutzer möchte ich alle unmittelbaren untergeordneten Komponenten für eine übergeordnete Komponenten-ID abrufen.	GSI1	<pre>ParentId = "&lt;ComponentId&gt;"</pre> <p>(Unmittelbare untergeordnete Elemente einer Komponente suchen.)</p>
Als Benutzer möchte ich eine rekursive Liste aller untergeordneten Komponenten für eine Komponenten-ID abrufen.	GSI1 oder GSI2	<pre>GSI1: ParentId = "&lt;ComponentId&gt;"</pre> <p>or</p> <pre>GSI2: GraphId = "&lt;TopLevelComponentId&gt;#N" AND BEGINS_WITH("Path", "&lt;PATH_OF_Component&gt;")</pre>

(Suche nach allen untergeordneten Komponenten mit Hilfe einer übergeordneten Komponente. Suche alle untergeordneten Komponenten mit Hilfe einer Komponente der mittleren Ebene).

Als Benutzer möchte ich die Vorfahren einer Komponente sehen.

Basistabelle

ComponentId = "<ComponentId>" Wählen Sie dann das Pfadattribut aus.

(Suche nach Vorfahren einer Komponente.)

Sie können auch ein Skript (Test) in jeder Programmiersprache implementieren, um DynamoDB direkt abzufragen und die Ergebnisse mit den erwarteten Ergebnissen zu vergleichen.

## Schritt 8: Überprüfen der Kostenschätzung

Überprüfen und verfeinern Sie die Kostenschätzung erneut. Darüber hinaus ist es eine bewährte Methode, sie mit Geschäfts-Stakeholdern zu validieren und die Genehmigung für den nächsten Schritt zu erhalten.

### Ziele

- Definieren Sie das Kapazitätsmodell und schätzen Sie die DynamoDB-Kosten, um die Kostenschätzung aus [Schritt 2](#) zu verfeinern.
- Holen Sie sich die endgültige finanzielle Genehmigung vom Geschäftsanalysten und den Stakeholdern.

### Prozess

- Der Datenbankingenieur identifiziert die Schätzung des Datenvolumens.
- Der Datenbankingenieur identifiziert die Datenübertragungsanforderungen.

- Der Datenbankingenieur definiert die erforderlichen Lese- und Schreibkapazitätseinheiten.
- Der Geschäftsanalyst entscheidet zwischen [On-Demand- und bereitgestellten Kapazitätsmodellen](#).
- Der Datenbankingenieur identifiziert die Notwendigkeit von [DynamoDB Auto Scaling](#).
- Der Datenbankingenieur gibt die Parameter in ein AWS Pricing Calculator.
- Der Datenbankingenieur präsentiert den Geschäfts-Stakeholdern die endgültige Preisschätzung.
- Geschäftsanalyst und Stakeholder genehmigen oder lehnen die Lösung ab.

## Schritt 9: Bereitstellen des Datenmodells

Für dieses spezifische Beispiel wurde die Bereitstellung des Modells mit [NoSQL Workbench durchgeführt](#), einer Anwendung für die moderne Datenbankentwicklung und den modernen Betrieb. Mit diesem Tool haben Sie die Möglichkeit, ein Datenmodell zu erstellen, Daten hochzuladen und direkt in Ihrem bereitgestellten AWS-Konto. Wenn Sie dieses Beispiel implementieren möchten, können Sie die folgende AWS CloudFormation Vorlage verwenden, die von NoSQL Workbench generiert wurde.

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  Components:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      KeySchema:
        - AttributeName: ComponentId
          KeyType: HASH
      AttributeDefinitions:
        - AttributeName: ComponentId
          AttributeType: S
        - AttributeName: ParentId
          AttributeType: S
        - AttributeName: GraphId
          AttributeType: S
        - AttributeName: Path
          AttributeType: S
      GlobalSecondaryIndexes:
        - IndexName: GS1
          KeySchema:
            - AttributeName: ParentId
              KeyType: HASH
            - AttributeName: ComponentId
```

```
    KeyType: RANGE
  Projection:
    ProjectionType: KEYS_ONLY
- IndexName: GSI2
  KeySchema:
    - AttributeName: GraphId
      KeyType: HASH
    - AttributeName: Path
      KeyType: RANGE
  Projection:
    ProjectionType: INCLUDE
  NonKeyAttributes:
    - ComponentId
BillingMode: PAY_PER_REQUEST
TableName: Components
```

# Weitere Ressourcen

## Weitere Informationen zu DynamoDB

- [DynamoDB-Preise](#)
- [DynamoDB-Dokumentation](#)
- [NoSQL-Design für DynamoDB](#)
- [Schreib-Sharding](#)
- [Lokale sekundäre Indizes \(LSIs\)](#)
- [Globale sekundäre Indizes \(GSIs\)](#)
- [Überladen von GSIs](#)
- [GSI-Sharding](#)
- [Verwenden von GSIs zum Erstellen eines Eventually-Consistent-Replikats](#)
- [Sparse-Indizes](#)
- [Materialisierte Aggregationsabfragen](#)
- [Zeitreihen-Designmuster](#)
- [Adjazenzlisten-Designmuster](#)
- [On-Demand- und bereitgestellte Kapazitätsmodelle](#)
- [DynamoDB Auto Scaling](#)
- [DynamoDB Time to Live \(TTL\)](#)
- [Modellieren von Gameplayer-Daten mit DynamoDB \(Lab\)](#)

## AWS-Services

- [AWS CloudFormation](#)
- [Amazon S3](#)

## Tools

- [AWS Pricing Calculator](#)
- [NoSQL Workbench für DynamoDB](#)
- [DynamoDB Local](#)

- [DynamoDB und AWS -SDKs](#)

## Bewährte Methoden

- [Bewährte Methoden für Design und Architektur mit DynamoDB](#) (DynamoDB-Dokumentation)
- [Bewährte Methoden für die Verwendung sekundärer Indizes](#) (DynamoDB-Dokumentation)
- [Bewährte Methoden zum Speichern großer Elemente und Attribute](#) (DynamoDB-Dokumentation)
- [Auswählen des richtigen DynamoDB-Partitionsschlüssels](#) (AWSDatenbank-Blog)
- [So entwerfen Sie globale sekundäre AmazonDynamoDBglobal-Indizes](#) (AWS-Datenbank-Blog)
- [Was sind Facetten in NoSQL Workbench für Amazon DynamoDB](#) (Medium-Website)

## AWS Allgemeine Ressourcen

- [AWS Website für Prescriptive Guidance](#)
- [AWS-Dokumentation](#)
- [AWS Allgemeine Referenz](#)

# Mitwirkende

Zu den Mitwirkenden dieses Handbuchs gehören:

- Camilo Gonzalez, Chef-Datenarchitekt, AWS
- Moinul Al-Mamun, Regional Big Data Architect, AWS
- S anti Segura, Professional Services Bol, AWS
- Satheish Kumar Chandraprakasam, Cloud Application Architect, AWS

# Dokumentverlauf

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

Änderung	Beschreibung	Datum
<a href="#">Es wurden ein Abschnitt Bewährte Methoden und ein Beispiel für die hierarchische Datenmodellierung hinzugefügt.</a>	Wir haben eine Zusammenfassung der <a href="#">bewährten DynamoDB-Methoden</a> und ein step-by-step Beispiel für das Entwerfen und Validieren eines <a href="#">hierarchischen Modells</a> hinzugefügt.	05. Dezember 2023
<a href="#">Erste Veröffentlichung</a>	—	26. Oktober 2020

# AWS Glossar zu präskriptiven Leitlinien

Im Folgenden finden Sie häufig verwendete Begriffe in Strategien, Leitfäden und Mustern von AWS Prescriptive Guidance. Um Einträge vorzuschlagen, verwenden Sie bitte den Link Feedback geben am Ende des Glossars.

## Zahlen

### 7 Rs

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

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

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

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

## A

### ABAC

Siehe [attributbasierte](#) Zugriffskontrolle.

### abstrahierte Dienste

Weitere Informationen finden Sie unter [Managed Services](#).

### ACID

Siehe [Atomarität, Konsistenz, Isolierung und Haltbarkeit](#).

### Aktiv-Aktiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden (mithilfe eines bidirektionalen Replikationstools oder dualer Schreibvorgänge) und beide Datenbanken Transaktionen von miteinander verbundenen Anwendungen während der Migration verarbeiten. Diese Methode unterstützt die Migration in kleinen, kontrollierten Batches, anstatt einen einmaligen Cutover zu erfordern. Es ist flexibler, erfordert aber mehr Arbeit als eine [aktiv-passive](#) Migration.

### Aktiv-Passiv-Migration

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

### Aggregatfunktion

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

## AI

Siehe [künstliche Intelligenz](#).

## AIOps

Siehe [Operationen mit künstlicher Intelligenz](#).

## Anonymisierung

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

## Anti-Muster

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

## Anwendungssteuerung

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

## Anwendungsportfolio

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

## künstliche Intelligenz (KI)

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

## Operationen mit künstlicher Intelligenz (AIOps)

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

## Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den

öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

### Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

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

### Attributbasierte Zugriffskontrolle (ABAC)

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

### autoritative Datenquelle

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

### Availability Zone

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

### AWS Framework für die Cloud-Einführung (AWS CAF)

Ein Framework mit Richtlinien und bewährten Verfahren, das Unternehmen bei der Entwicklung eines effizienten und effektiven Plans für den erfolgreichen Umstieg auf die Cloud unterstützt. AWS CAF unterteilt die Leitlinien in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Unternehmen, Mitarbeiter, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Aus dieser Perspektive bietet AWS CAF Leitlinien für Personalentwicklung, Schulung und Kommunikation, um das Unternehmen auf eine erfolgreiche Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der [AWS -CAF-Webseite](#) und dem [AWS -CAF-Whitepaper](#).

## AWS Workload-Qualifizierungsrahmen (AWS WQF)

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

## B

### schlechter Bot

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

### BCP

Siehe [Planung der Geschäftskontinuität](#).

### Verhaltensdiagramm

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

### Big-Endian-System

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

### Binäre Klassifikation

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

### Bloom-Filter

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

### Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue

Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

## Bot

Eine Softwareanwendung, die automatisierte Aufgaben über das Internet ausführt und menschliche Aktivitäten oder Interaktionen simuliert. Manche Bots sind nützlich oder nützlich, wie z. B. Webcrawler, die Informationen im Internet indexieren. Einige andere Bots, die als bösartige Bots bezeichnet werden, sollen Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen.

## Botnetz

Netzwerke von [Bots](#), die mit [Malware](#) infiziert sind und unter der Kontrolle einer einzigen Partei stehen, die als Bot-Herder oder Bot-Operator bezeichnet wird. Botnetze sind der bekannteste Mechanismus zur Skalierung von Bots und ihrer Wirkung.

## branch

Ein containerisierter Bereich eines Code-Repositorys. Der erste Zweig, der in einem Repository erstellt wurde, ist der Hauptzweig. Sie können einen neuen Zweig aus einem vorhandenen Zweig erstellen und dann Feature entwickeln oder Fehler in dem neuen Zweig beheben. Ein Zweig, den Sie erstellen, um ein Feature zu erstellen, wird allgemein als Feature-Zweig bezeichnet. Wenn das Feature zur Veröffentlichung bereit ist, führen Sie den Feature-Zweig wieder mit dem Hauptzweig zusammen. Weitere Informationen finden Sie unter [Über Branches](#) (GitHub Dokumentation).

## Zugang durch Glasbruch

Unter außergewöhnlichen Umständen und im Rahmen eines genehmigten Verfahrens ist dies eine schnelle Methode für einen Benutzer, auf einen Bereich zuzugreifen AWS-Konto, für den er normalerweise keine Zugriffsrechte besitzt. Weitere Informationen finden Sie unter dem Indikator [Implementation break-glass procedures](#) in den AWS Well-Architected-Leitlinien.

## Brownfield-Strategie

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

## Puffer-Cache

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

## Geschäftsfähigkeit

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

## Planung der Geschäftskontinuität (BCP)

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

## C

### CAF

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

### Bereitstellung auf Kanaren

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

### CCoE

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

### CDC

Siehe [Erfassung von Änderungsdaten](#).

### Erfassung von Datenänderungen (CDC)

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

### Chaos-Technik

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

## CI/CD

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

## Klassifizierung

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

## clientseitige Verschlüsselung

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

## Cloud-Kompetenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte Transformationen. Weitere Informationen finden Sie in den [CCoE-Beiträgen](#) im AWS Cloud Enterprise Strategy Blog.

## Cloud Computing

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

## Cloud-Betriebsmodell

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

## Phasen der Einführung der Cloud

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

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

- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

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

## CMDB

Siehe [Datenbank für das Konfigurationsmanagement](#).

## Code-Repository

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

## Kalter Cache

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

## Kalte Daten

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

## Computer Vision (CV)

Ein Bereich der [KI](#), der maschinelles Lernen nutzt, um Informationen aus visuellen Formaten wie digitalen Bildern und Videos zu analysieren und zu extrahieren. AWS Panorama bietet beispielsweise Geräte an, die CV zu lokalen Kameranetzwerken hinzufügen, und Amazon SageMaker stellt Bildverarbeitungsalgorithmen für CV bereit.

## Drift in der Konfiguration

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

## Verwaltung der Datenbankkonfiguration (CMDB)

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

## Konformitätspaket

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

## Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

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

## CV

Siehe [Computer Vision](#).

## D

### Daten im Ruhezustand

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

### Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil

der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter [Datenklassifizierung](#).

### Datendrift

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

### Daten während der Übertragung

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

### Datennetz

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

### Datenminimierung

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

### Datenperimeter

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

### Vorverarbeitung der Daten

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

### Herkunft der Daten

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

### betreffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

## Data Warehouse

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

## Datenbankdefinitionssprache (DDL)

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

## Datenbankmanipulationssprache (DML)

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

## DDL

Siehe [Datenbankdefinitionssprache](#).

## Deep-Ensemble

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

## Deep Learning

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

## defense-in-depth

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

## delegierter Administrator

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

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

## Bereitstellung

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

## Entwicklungsumgebung

Siehe [Umgebung](#).

## Detektivische Kontrolle

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

## Abbildung des Wertstroms in der Entwicklung (DVSM)

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

## digitaler Zwilling

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

## Maßtabelle

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

## Katastrophe

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

## Notfallwiederherstellung (DR)

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

## DML

Siehe Sprache zur [Datenbankmanipulation](#).

## Domainorientiertes Design

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

## DR

Siehe [Disaster Recovery](#).

## Erkennung von Driften

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

## DVSM

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

# E

## EDA

Siehe [explorative Datenanalyse](#).

## Edge-Computing

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

## Verschlüsselung

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

## Verschlüsselungsschlüssel

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

## Endianismus

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

## Endpunkt

[Siehe](#) Service-Endpunkt.

## Endpunkt-Services

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

## Unternehmensressourcenplanung (ERP)

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

## Envelope-Verschlüsselung

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

## Umgebung

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

- **Entwicklungsumgebung** – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.
- **Niedrigere Umgebungen** – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.
- **Produktionsumgebung** – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD-Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- **Höhere Umgebungen** – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

## Epics

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

## ERP

Siehe [Enterprise Resource Planning](#).

## Explorative Datenanalyse (EDA)

Der Prozess der Analyse eines Datensatzes, um seine Hauptmerkmale zu verstehen. Sie sammeln oder aggregieren Daten und führen dann erste Untersuchungen durch, um Muster zu

finden, Anomalien zu erkennen und Annahmen zu überprüfen. EDA wird durchgeführt, indem zusammenfassende Statistiken berechnet und Datenvisualisierungen erstellt werden.

## F

### Faktentabelle

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

### schnell scheitern

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

### Grenze zur Fehlerisolierung

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

### Feature-Zweig

Siehe [Zweig](#).

### Features

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

### Bedeutung der Feature

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

## Featuretransformation

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

## FGAC

Weitere Informationen finden Sie unter [detaillierter Zugriffskontrolle](#).

### Feinkörnige Zugriffskontrolle (FGAC)

Die Verwendung mehrerer Bedingungen, um eine Zugriffsanfrage zuzulassen oder abzulehnen.

## Flash-Cut-Migration

Eine Datenbankmigrationsmethode, bei der eine kontinuierliche Datenreplikation durch [Erfassung von Änderungsdaten](#) verwendet wird, um Daten in kürzester Zeit zu migrieren, anstatt einen schrittweisen Ansatz zu verwenden. Ziel ist es, Ausfallzeiten auf ein Minimum zu beschränken.

## G

### Geoblocking

Siehe [geografische Einschränkungen](#).

### Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden, um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in [der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte](#). CloudFront

### Gitflow-Workflow

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

## Greenfield-Strategie

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

## Integritätsschutz

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

# H

## HEKTAR

Siehe [Hochverfügbarkeit](#).

## Heterogene Datenbankmigration

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

## hohe Verfügbarkeit (HA)

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

## historische Modernisierung

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

## Homogene Datenbankmigration

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

## heiße Daten

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

## Hotfix

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

## Hypercare-Phase

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

## I

## IaC

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

## Identitätsbasierte Richtlinie

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

## Leerlaufanwendung

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

## IIoT

Siehe [Industrielles Internet der Dinge](#).

## unveränderliche Infrastruktur

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

## Eingehende (ingress) VPC

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

## Inkrementelle Migration

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

## Industrie 4.0

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

## Infrastruktur

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

## Infrastructure as Code (IaC)

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

## Industrielles Internet der Dinge (IIoT)

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

## Inspektions-VPC

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

## Internet of Things (IoT)

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

## Interpretierbarkeit

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

## IoT

[Siehe Internet der Dinge.](#)

## IT information library (ITIL, IT-Informationsbibliothek)

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

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

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

## BIS

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

## ITSM

Siehe [IT-Servicemanagement](#).

## L

### Labelbasierte Zugangskontrolle (LBAC)

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

### Landing Zone

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

### Große Migration

Eine Migration von 300 oder mehr Servern.

## SCHWARZ

Weitere Informationen finden Sie unter [Label-basierte Zugriffskontrolle](#).

### Geringste Berechtigung

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

## Lift and Shift

Siehe [7 Rs](#).

## Little-Endian-System

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

## Niedrigere Umgebungen

[Siehe Umwelt](#).

# M

## Machine Learning (ML)

Eine Art künstlicher Intelligenz, die Algorithmen und Techniken zur Mustererkennung und zum Lernen verwendet. ML analysiert aufgezeichnete Daten, wie z. B. Daten aus dem Internet der Dinge (IoT), und lernt daraus, um ein statistisches Modell auf der Grundlage von Mustern zu erstellen. Weitere Informationen finden Sie unter [Machine Learning](#).

## Hauptzweig

Siehe [Filiale](#).

## Malware

Software, die entwickelt wurde, um die Computersicherheit oder den Datenschutz zu gefährden. Malware kann Computersysteme stören, vertrauliche Informationen durchsickern lassen oder sich unbefugten Zugriff verschaffen. Beispiele für Malware sind Viren, Würmer, Ransomware, Trojaner, Spyware und Keylogger.

## verwaltete Dienste

AWS-Services für die die Infrastrukturebene, das Betriebssystem und die Plattformen AWS betrieben werden, und Sie greifen auf die Endgeräte zu, um Daten zu speichern und abzurufen. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB sind Beispiele für Managed Services. Diese werden auch als abstrakte Dienste bezeichnet.

## Manufacturing Execution System (MES)

Ein Softwaresystem zur Nachverfolgung, Überwachung, Dokumentation und Steuerung von Produktionsprozessen, bei denen Rohstoffe in der Fertigung zu fertigen Produkten umgewandelt werden.

## MAP

Siehe [Migration Acceleration Program](#).

## Mechanismus

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

## Mitgliedskonto

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

## DURCHEINANDER

Siehe [Manufacturing Execution System](#).

## Message Queuing-Telemetrietransport (MQTT)

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

## Microservice

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

## Microservices-Architekturen

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

## Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

### Migration in großem Maßstab

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

### Migrationsfabrik

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

### Migrationsmetadaten

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

### Migrationsmuster

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

## Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung, Migrationspriorisierung und Wellenplanung). Das [MPA-Tool](#) (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

## Migration Readiness Assessment (MRA)

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

## Migrationsstrategie

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

## ML

[Siehe maschinelles Lernen.](#)

## Modernisierung

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

## Bewertung der Modernisierungsfähigkeit

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

## Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter [Zerlegen von Monolithen in Microservices](#).

## MPA

Siehe [Bewertung des Migrationsportfolios](#).

## MQTT

Siehe [Message Queuing-Telemetrietransport](#).

## Mehrklassen-Klassifizierung

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

## veränderbare Infrastruktur

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

## O

### OAC

[Weitere Informationen finden Sie unter Origin Access Control](#).

### OAI

Siehe [Zugriffsidentität von Origin](#).

### COM

Siehe [organisatorisches Change-Management](#).

## Offline-Migration

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

OI

Siehe [Betriebsintegration](#).

OLA

Siehe Vereinbarung auf [operativer Ebene](#).

## Online-Migration

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

OPC-UA

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

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

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

## Vereinbarung auf Betriebsebene (OLA)

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

## Überprüfung der Betriebsbereitschaft (ORR)

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

## Betriebstechnologie (OT)

Hardware- und Softwaresysteme, die mit der physischen Umgebung zusammenarbeiten, um industrielle Abläufe, Ausrüstung und Infrastruktur zu steuern. In der Fertigung ist die Integration

von OT- und Informationstechnologie (IT) -Systemen ein zentraler Schwerpunkt der [Industrie 4.0-Transformationen](#).

### Betriebsintegration (OI)

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

### Organisationspfad

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

### Organisatorisches Veränderungsmanagement (OCM)

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

### Ursprungszugriffskontrolle (OAC)

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

### Ursprungszugriffsidentität (OAI)

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

## ODER

Siehe [Überprüfung der Betriebsbereitschaft](#).

## NICHT

Siehe [Betriebstechnologie](#).

## Ausgehende (egress) VPC

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

## P

### Berechtigungsgrenze

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

### persönlich identifizierbare Informationen (PII)

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

### Personenbezogene Daten

Siehe [persönlich identifizierbare Informationen](#).

### Playbook

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

## PLC

Siehe [programmierbare Logiksteuerung](#).

## PLM

Siehe [Produktlebenszyklusmanagement](#).

### policy

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

### Polyglotte Beharrlichkeit

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

### Portfoliobewertung

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

### predicate

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

### Prädikat Pushdown

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

### Präventive Kontrolle

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

## Prinzipal

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

## Datenschutz durch Design

Ein Ansatz in der Systemtechnik, der den Datenschutz während des gesamten Engineering-Prozesses berücksichtigt.

## Privat gehostete Zonen

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

## proaktive Steuerung

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

## Produktlebenszyklusmanagement (PLM)

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

## Produktionsumgebung

Siehe [Umgebung](#).

## Speicherprogrammierbare Steuerung (SPS)

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

## Pseudonymisierung

Der Prozess, bei dem persönliche Identifikatoren in einem Datensatz durch Platzhalterwerte ersetzt werden. Pseudonymisierung kann zum Schutz der Privatsphäre beitragen.

Pseudonymisierte Daten gelten weiterhin als personenbezogene Daten.

## veröffentlichen/abonnieren (pub/sub)

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

## Q

### Abfrageplan

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

### Abfrageplanregression

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

## R

### RACI-Matrix

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

### Ransomware

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

### RASCI-Matrix

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

## RCAC

Siehe [Zugriffskontrolle für Zeilen und Spalten](#).

## Read Replica

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

## neu strukturieren

Siehe [7 Rs](#).

## Recovery Point Objective (RPO)

Die maximal zulässige Zeitspanne seit dem letzten Datenwiederherstellungspunkt. Dies bestimmt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Betriebsunterbrechung angesehen wird.

## Wiederherstellungszeitziel (RTO)

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

## Refaktorisierung

Siehe [7 Rs](#).

## Region

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

## Regression

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

## rehosten

Siehe [7 Rs](#).

## Veröffentlichung

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

umziehen

Siehe [7 Rs.](#)

neue Plattform

Siehe [7 Rs.](#)

Rückkauf

Siehe [7 Rs.](#)

Ausfallsicherheit

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

Ressourcenbasierte Richtlinie

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

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

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

Reaktive Kontrolle

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

## Beibehaltung

Siehe [7 Rs](#).

## zurückziehen

Siehe [7 Rs](#).

## Drehung

Der Vorgang, bei dem ein [Geheimnis](#) regelmäßig aktualisiert wird, um es einem Angreifer zu erschweren, auf die Anmeldeinformationen zuzugreifen.

## Zugriffskontrolle für Zeilen und Spalten (RCAC)

Die Verwendung einfacher, flexibler SQL-Ausdrücke mit definierten Zugriffsregeln. RCAC besteht aus Zeilenberechtigungen und Spaltenmasken.

## RPO

Siehe [Recovery Point Objective](#).

## RTO

Siehe [Ziel der Wiederherstellungszeit](#).

## Runbook

Eine Reihe manueller oder automatisierter Verfahren, die zur Ausführung einer bestimmten Aufgabe erforderlich sind. Diese sind in der Regel darauf ausgelegt, sich wiederholende Operationen oder Verfahren mit hohen Fehlerquoten zu rationalisieren.

# S

## SAML 2.0

Ein offener Standard, den viele Identitätsanbieter (IdPs) verwenden. Diese Funktion ermöglicht föderiertes Single Sign-On (SSO), sodass sich Benutzer bei den API-Vorgängen anmelden AWS Management Console oder die AWS API-Operationen aufrufen können, ohne dass Sie einen Benutzer in IAM für alle in Ihrer Organisation erstellen müssen. Weitere Informationen zum SAML-2.0.-basierten Verbund finden Sie unter [Über den SAML-2.0-basierten Verbund](#) in der IAM-Dokumentation.

## SCADA

Siehe [Aufsichtskontrolle und Datenerfassung](#).

## SCP

Siehe [Richtlinie zur Dienstkontrolle](#).

## Secret

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

## Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: [präventiv](#), [detektiv](#), [reaktionsschnell](#) und [proaktiv](#).

## Härtung der Sicherheit

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

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

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

## Automatisierung von Sicherheitsreaktionen

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

## Serverseitige Verschlüsselung

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

## Service-Kontrollrichtlinie (SCP)

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

## Service-Endpunkt

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

## Service Level Agreement (SLA)

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

## Service-Level-Indikator (SLI)

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

## Service-Level-Ziel (SLO)

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

## Modell der geteilten Verantwortung

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

## SIEM

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

## Single Point of Failure (SPOF)

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

## SLA

Siehe [Service Level Agreement](#).

## SLI

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

## ALSO

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

## split-and-seed Modell

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

## SPOTTEN

Siehe [Single Point of Failure](#).

## Sternschema

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

## Strangler-Fig-Muster

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

unter [Schrittweises Modernisieren älterer Microsoft ASP.NET \(ASMX\)-Webservices mithilfe von Containern und Amazon API Gateway](#).

## Subnetz

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

## Aufsichtskontrolle und Datenerfassung (SCADA)

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

## Symmetrische Verschlüsselung

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

## synthetisches Testen

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

# T

## tags

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

## Zielvariable

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

## Aufgabenliste

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

## Testumgebungen

[Siehe Umgebung.](#)

## Training

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

## Transit-Gateway

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

## Stammbasierter Workflow

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

## Vertrauenswürdiger Zugriff

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

## Optimieren

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

## Zwei-Pizzen-Team

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

# U

## Unsicherheit

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

## undifferenzierte Aufgaben

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

## höhere Umgebungen

Siehe [Umgebung](#).

# V

## Vacuuming

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

## Versionskontrolle

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

## VPC-Peering

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

## Schwachstelle

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

## W

### Warmer Cache

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

### warme Daten

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

### Fensterfunktion

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

### Workload

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

### Workstream

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

### WURM

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

### WQF

Weitere Informationen finden Sie unter [AWS Workload Qualification Framework](#).

## einmal schreiben, viele lesen (WORM)

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

## Z

### Zero-Day-Exploit

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

### Zero-Day-Sicherheitslücke

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

### Zombie-Anwendung

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

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