



Automatisieren Sie Ihre DR-Lösung für relationale Datenbanken auf AWS

# AWS Prescriptive



# AWS Prescriptive: Automatisieren Sie Ihre DR-Lösung für relationale Datenbanken auf AWS

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
Ziele .....	1
Übersicht .....	3
Der Failover-Prozess .....	4
DR Orchestrator FAILOVER-Architektur .....	4
Der Failback-Prozess .....	5
DR Orchestrator FAILBACK-Architektur .....	5
DR Orchestrator FAILOVER .....	7
Einzelheiten zu den Parametern .....	8
DR Orchestrator FAILBACK .....	9
Aurora-Zustände und Step Functions Functions-Zustandsmaschinen .....	12
Stetiger Zustand .....	12
Status des Ereignisses .....	12
Umstellung .....	13
Manuelles ungeplantes Failover .....	18
Failback .....	22
Stellen Sie die Lösung bereit .....	31
FAQ .....	33
Was sind die RPO und was kann RTO ich mit diesem Ansatz erreichen? .....	33
Ist die Verwendung von AWS CloudFormation Exportvariablen obligatorisch? .....	33
Kann ich den DR FAILOVER Orchestrator-Workflow verwenden, um ein Failover für mehr als eine AWS Datenbank durchzuführen? .....	33
Wie kann ich den InvalidParameterCombination Fehler vermeiden, wenn ich die DR FAILBACK Orchestrator-Zustandsmaschine für Amazon RDS ausführe? .....	34
Nächste Schritte .....	35
Ressourcen .....	36
Dokumentverlauf .....	37
Glossar .....	38
# .....	38
A .....	39
B .....	42
C .....	44
D .....	48
E .....	52

F .....	54
G .....	56
H .....	57
I .....	58
L .....	61
M .....	62
O .....	66
P .....	69
Q .....	72
R .....	72
S .....	75
T .....	79
U .....	81
V .....	81
W .....	82
Z .....	83
.....	lxxxiv

# Automatisieren Sie Ihre DR-Lösung für relationale Datenbanken auf AWS

Jitendra Kumar, Oliver Francis und Pavithra Balasubramanian, Amazon Web Services (AWS)

Mai 2024 ([Geschichte der Dokumente](#))

Bei der Planung einer [Disaster Recovery \(DR\) -Strategie](#) für Amazon Web Services (AWS) können Sie eine DR-Lösung für die Datenbanken in Ihrem Unternehmen implementieren. Sie können die DR-Lösung automatisieren, indem Sie eine ereignisgesteuerte Architektur für den Failover der Datenbank-Instances konfigurieren. Zu den Vorteilen dieses Ansatzes gehören:

- Reduzierung menschlicher Fehler, die durch manuelle Koordination verursacht werden
- In einigen Fällen wird das Ziel für die kürzest mögliche Wiederherstellungszeit (RTO) erreicht
- Erleichterung wiederholbarer Tests Ihrer DR-Lösung

Ohne Automatisierung würde ein Ausfall in Ihrer Primäranlage Ressourcen in Ihrem Unternehmen erfordern AWS-Region, um Schritte in einem Runbook manuell durchzuführen. Diese Schritte können mehrere Prozesse umfassen, z. B. das Heraufstufen von Amazon Relational Database Service (AmazonRDS) -Read Replicas in der Sekundärdatenbank AWS-Region, sodass sie als neue primäre Datenbank-Instance dienen. AWS bietet [AWS SDKPython \(Boto3\)](#) APIs zur Automatisierung dieser Aktionen, die die Bausteine einer DR-Lösung für die Bedürfnisse Ihres Unternehmens bilden können.

In diesem Leitfaden wird ein zentralisierter, manifestgesteuerter DR-Orchestrator beschrieben, der Ihnen helfen kann, eine DR-Lösung für AWS Datenbanken innerhalb derselben zu entwickeln. AWS-Regionen AWS-Konto Die Entscheidung, einen Failover durchzuführen, würde immer noch menschliches Eingreifen erfordern und ist nicht automatisierbar.

Die Zielgruppen dieses Leitfadens sind Anwendungsarchitekten, Infrastrukturarchitekten, Datenbankarchitekten und Datenbankberater.

## Ziele

Durch die Implementierung der DR Orchestrator Framework-Lösung können Sie die folgenden Ergebnisse erzielen:

- Reduzieren Sie manuelle Fehler bei einem Ausfall — Die manuelle Ausführung von DR-Prozessen kann fehleranfällig sein. Die Automatisierung von Aktionen während der Phasen einer DR-Aktivität reduziert die Anzahl manueller Fehler und trägt zu einem reibungsloseren Failover oder Failback bei.
- DR in großem Umfang verwalten — Wenn Ihr Unternehmen über eine große Flotte von Datenbanken verfügt, hilft Ihnen Building Automation for DR dabei, Ihre DR-Lösung zu skalieren. Sie können Ihre AWS Datenbank-DR-Lösung mithilfe des DR Orchestrator Framework automatisieren. DR Orchestrator automatisiert die Schritte für Failover oder Failback von AWS Datenbanken mithilfe einer Parameterdatei im Format. JSON

# Überblick über das DR Orchestrator Framework

DR Orchestrator Framework bietet eine Ein-Klick-Lösung zur Orchestrierung und Automatisierung regionsübergreifender DR für Datenbanken. AWS Es verwendet [AWS Step Functions](#) und [AWS Lambda](#) führt die erforderlichen Schritte während des Failovers und des Failbacks aus. Die Step Functions Zustandsmaschinen bilden die Grundlage für die Entscheidungsfindung innerhalb des Orchestrator-Designs. Die API Operationen zur Durchführung von Failover- oder Failback-Aktionen sind in Lambda-Funktionen codiert, die von der Zustandsmaschine aus aufgerufen werden. Die Lambda-Funktionen werden ausgeführt [AWS SDK for Python \(Boto3\)](#) APIs, um mit AWS Datenbanken zu interagieren.

Das DR Orchestrator Framework enthält zwei Hauptzustandsmaschinen, die den Failover- und Failback-Phasen entsprechen.

Bei Amazon RDS wird in der Failover-Phase eine regionsübergreifende RDS Read Replica in eine eigenständige DB-Instance umgewandelt. Bei Amazon Aurora ist der Writer-Node nicht verfügbar, wenn die primäre Region während eines seltenen, unerwarteten Ausfalls ausgefallen ist. Die Replikation zwischen dem Writer-Knoten und den sekundären Clustern wird gestoppt. Sie müssen den sekundären Cluster von der globalen Datenbank trennen und ihn zu einem eigenständigen Cluster heraufstufen. Anwendungen können eine Verbindung herstellen und Schreibdatenverkehr an den eigenständigen Cluster senden. Sie können denselben Prozess verwenden, um vom primären DB-Cluster der globalen Datenbank zu den sekundären Regionen zu [wechseln](#). Verwenden Sie diesen Ansatz für kontrollierte Szenarien wie die folgenden:

- Betriebliche Wartung
- Geplante Betriebsabläufe
- Förderung eines sekundären Amazon ElastiCache (RedisOSS) -Clusters als neuen primären Cluster

In der Failback-Phase wird die Live-Replikation von Daten zwischen einer primären Live-Region und einer neuen sekundären Region eingerichtet.

Es ist wichtig zu verstehen, dass DR Orchestrator nur für Datenbanken gilt. Alle Anwendungen, die auf diese Datenbanken verweisen und sich in derselben Region befinden, benötigen möglicherweise eine separate Tandem-Failover-Lösung. Nach dem Failover der Datenbanken in die sekundäre

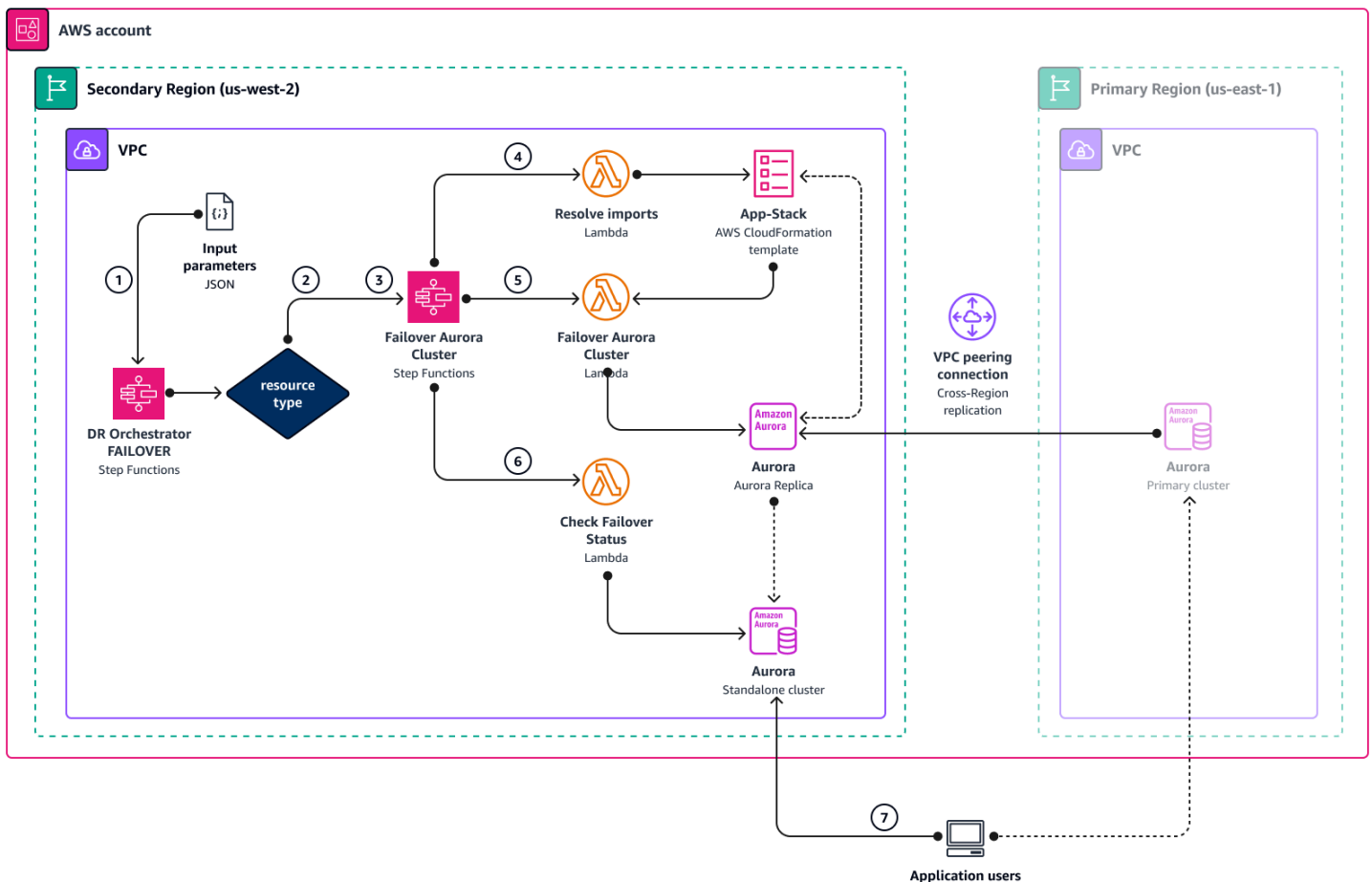
Region müssen die Anwendungen aktualisiert werden, um eine Verbindung zu den neuen Datenbankinstanzen herzustellen, die als Datenquelle dienen.

## Der Failover-Prozess

Führen Sie die DR Orchestrator FAILOVER Zustandsmaschine aus, um einen Failover durchzuführen. Zu diesem Zeitpunkt ist in der sekundären Region bereits eine sekundäre Datenbank vorhanden, entweder als Read Replica (AmazonRDS) oder als sekundärer Cluster (Amazon Aurora). Wenn Sie die DR Orchestrator FAILOVER Zustandsmaschine ausführen, wird die sekundäre Datenbank zur primären Datenbank heraufgestuft.

## DR Orchestrator FAILOVER-Architektur

Das folgende Diagramm zeigt die Konzepte des Failover-Prozesses für Amazon Aurora bei Verwendung von DR Orchestrator. Amazon Aurora und Amazon ElastiCache verwenden denselben Workflow, jedoch mit unterschiedlichen Zustandsmaschinen und Lambda-Funktionen.





1. Die `DR Orchestrator FAILOVER` Zustandsmaschine liest die JSON Eingabeparameter.
2. Basierend auf dem `resourceType` Parameter ruft die Zustandsmaschine andere Zustandsmaschinen auf: `Promote RDS Read ReplicaFailover Aurora Cluster`, `oderFailover ElastiCache`. Wenn in der Eingabe mehr als eine Ressource übergeben wird, laufen diese Zustandsmaschinen parallel.
3. Die `Failover Aurora Cluster` Zustandsmaschine ruft Lambda-Funktionen in jedem der folgenden drei Schritte auf.
4. Die `Resolve imports` Lambda-Funktion wird `"! import <export-variable-name>"` mit den tatsächlichen Werten aus der App-Stack AWS CloudFormation Vorlage aufgelöst.
5. Die **Failover Aurora Cluster** Lambda-Funktion bewirbt die Read Replica als eigenständige DB-Instance.
6. Die **Check Failover Status** Lambda-Funktion überprüft den Status der beworbenen DB-Instance. Wenn der Status lautet `AVAILABLE`, sendet die Lambda-Funktion ein Erfolgstoken zurück an die aufrufende Zustandsmaschine und schließt den Vorgang ab.
7. Sie können Ihre Anwendungen zur eigenständigen Datenbank in der DR-Region (`us-west-2`) umleiten, die jetzt die primäre Datenbank ist.

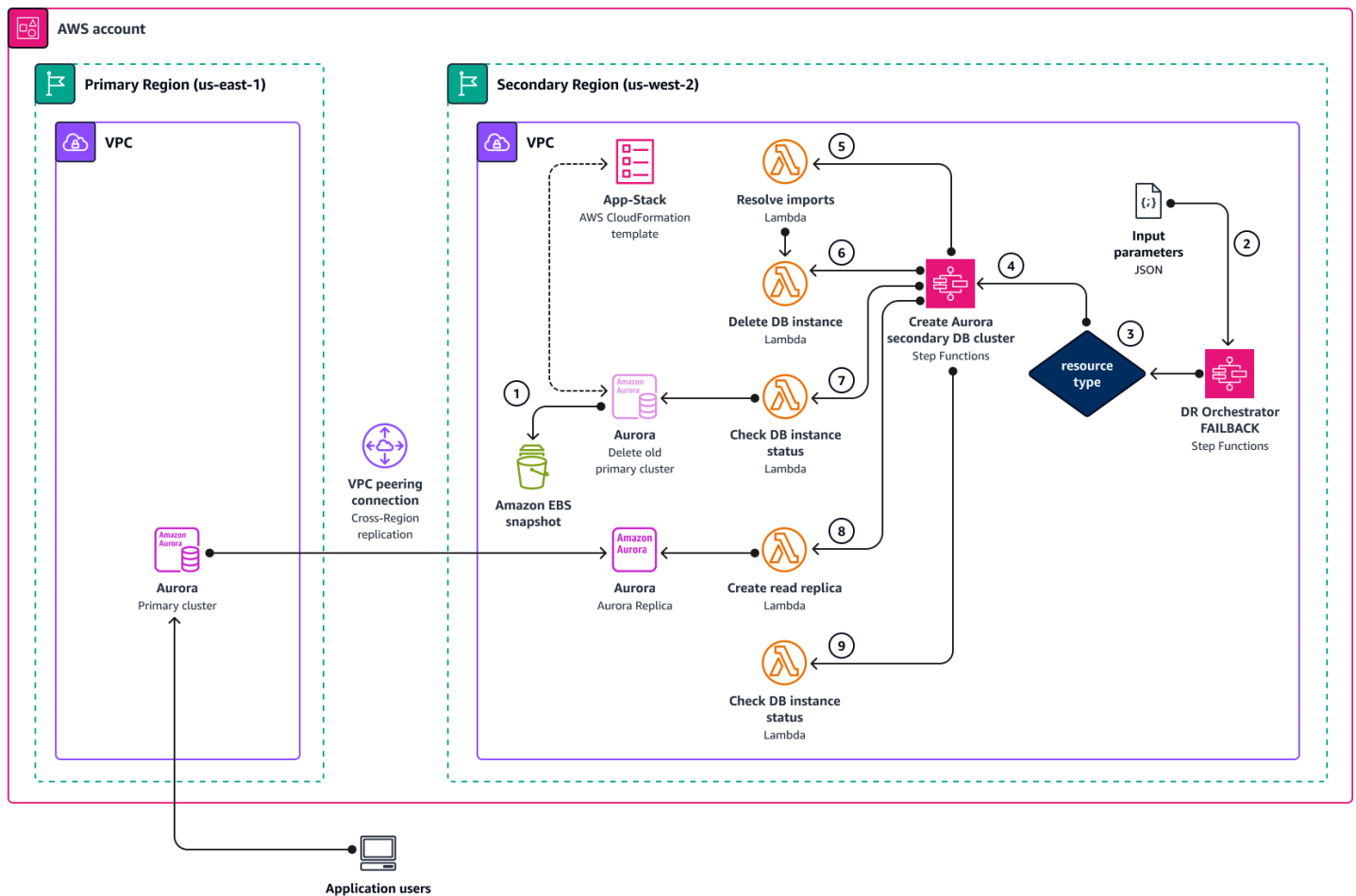
## Der Failback-Prozess

Nachdem Ihre frühere primäre Region (`us-east-1`) wieder aktiv ist, können Sie zu ihr zurückkehren, sodass die Datenbank in wieder zur primären Region `us-east-1` wird. Um das Failback zu starten, führen Sie die `DR Orchestrator FAILBACK` Zustandsmaschine aus. Wie der Name schon sagt, beginnt dieser Zustandsmaschine damit, Änderungen in Ihrer neuen primären Region (`us-west-2`) zurück in die frühere primäre Region (`us-east-1`) zu replizieren, die als aktuelle sekundäre Region fungiert.

Nachdem die Replikation zwischen den beiden Regionen eingerichtet wurde, können Sie das Failback einleiten. Um ein Failback durchzuführen und zu Ihrer ursprünglichen primären Region (`us-east-1`) zurückzukehren, führen Sie den `DR Orchestrator FAILOVER` Zustandsmaschine in der aktuellen sekundären Region (`us-east-1`) aus, um sie zur primären Region hochzustufen.

## DR Orchestrator FAILBACK-Architektur

Das folgende Diagramm zeigt die Konzepte des Failback-Prozesses für Amazon Aurora bei Verwendung von DR Orchestrator.



1. Bevor Sie mit dem Failback beginnen, erstellen Sie einen manuellen DB-Snapshot, den Sie bei der Ursachenanalyse verwenden können (). RCA

Deaktivieren Sie außerdem das DeletionProtection für den Aurora-Cluster in der vorherigen primären Region (us-east-1).

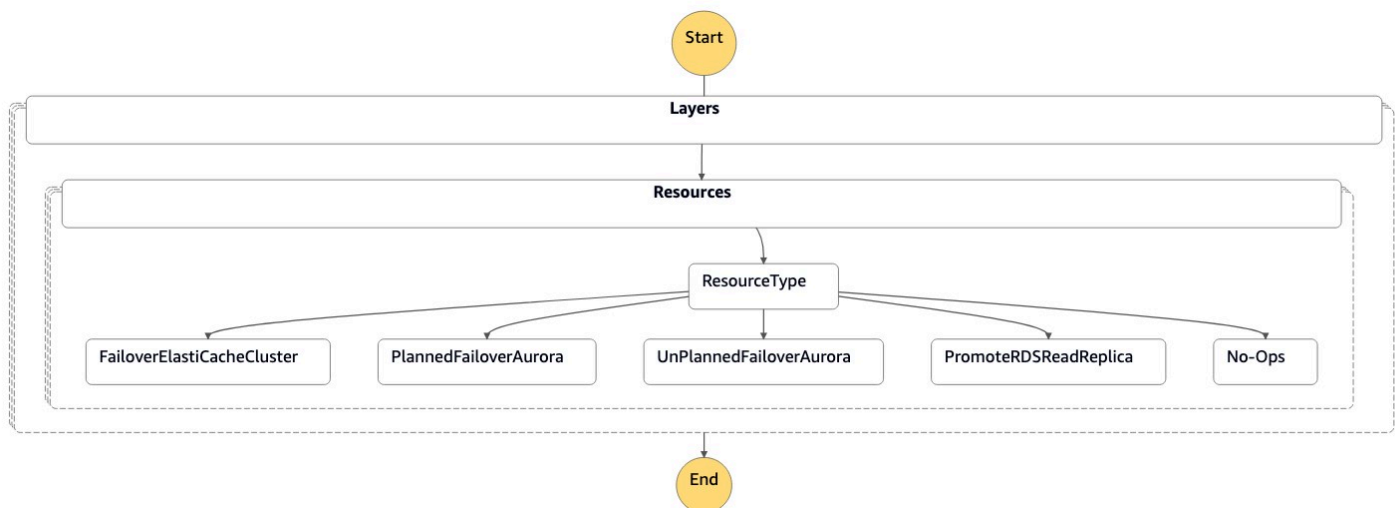
2. Die DR Orchestrator FAILBACK Zustandsmaschine liest die JSON Eingabeparameter.
3. Basierend auf dem resourceType ruft die DR Orchestrator FAILBACK Zustandsmaschine die Create Aurora Secondary DB Cluster Zustandsmaschine auf.
4. Die Create Aurora Secondary DB Cluster Zustandsmaschine ruft Lambda-Funktionen in jedem der folgenden fünf Schritte auf.
5. Die Resolve import Lambda-Funktion wird "! import <export-variable-name>" mit den tatsächlichen Werten aus der App-Stack CloudFormation Vorlage aufgelöst.
6. Die Delete DB Instance Lambda-Funktion löscht die frühere Primärinstanz.

7. Die Check DB instance status Lambda-Funktion überprüft die, Delete DB Instance status bis die Datenbank gelöscht wird.
8. Die Create Read Replica Lambda-Funktion erstellt eine Read Replica in der sekundären Region aus der DB-Instance, die sich in der neuen primären Region befindet.
9. Die Check DB instance status Lambda-Funktion überprüft den Status der Read Replica-DB-Instance. Wenn der Status lautet AVAILABLE, sendet die Lambda-Funktion ein Erfolgstoken zurück an die aufrufende Zustandsmaschine, die abgeschlossen ist.

## DR Orchestrator FAILOVER

Verwenden Sie die DR Orchestrator FAILOVER Zustandsmaschine im DR-Ereignis, wenn die primäre Region (us-east-1) ausgefallen ist, oder bei geplanten Ereignissen, z. B. bei Wartungsarbeiten.

Die Funktion kann aufgerufen werden, um ein Failover einzelner oder mehrerer Datenbanken parallel durchzuführen.



Die Zustandsmaschine akzeptiert Parameter in dem JSON Format, das im folgenden Code gezeigt wird:

```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
```

```
{
  "resourceType": "PromoteRDSReadReplica",
  "resourceName": "Promote RDS MySQL Read Replica",
  "parameters": {
    "RDSInstanceIdentifier": "!Import rds-mysql-instance-identifier",
    "TargetClusterIdentifier": "!Import rds-mysql-instance-global-arn"
  }
},
{
  "resourceType": "FailoverElastiCacheCluster",
  "resourceName": "Failover ElastiCache Cluster",
  "parameters": {
    "GlobalReplicationGroupId": "!Import demo-redis-cluster-global-replication-
group-id",
    "TargetRegion": "!Import demo-redis-cluster-target-region",
    "TargetReplicationGroupId": "!Import demo-redis-cluster-target-replication-
group-id"
  }
}
]
```

## Einzelheiten zu den Parametern

Die folgende Tabelle zeigt die von der DR Orchestrator FAILOVER Zustandsmaschine verwendeten Parameter.

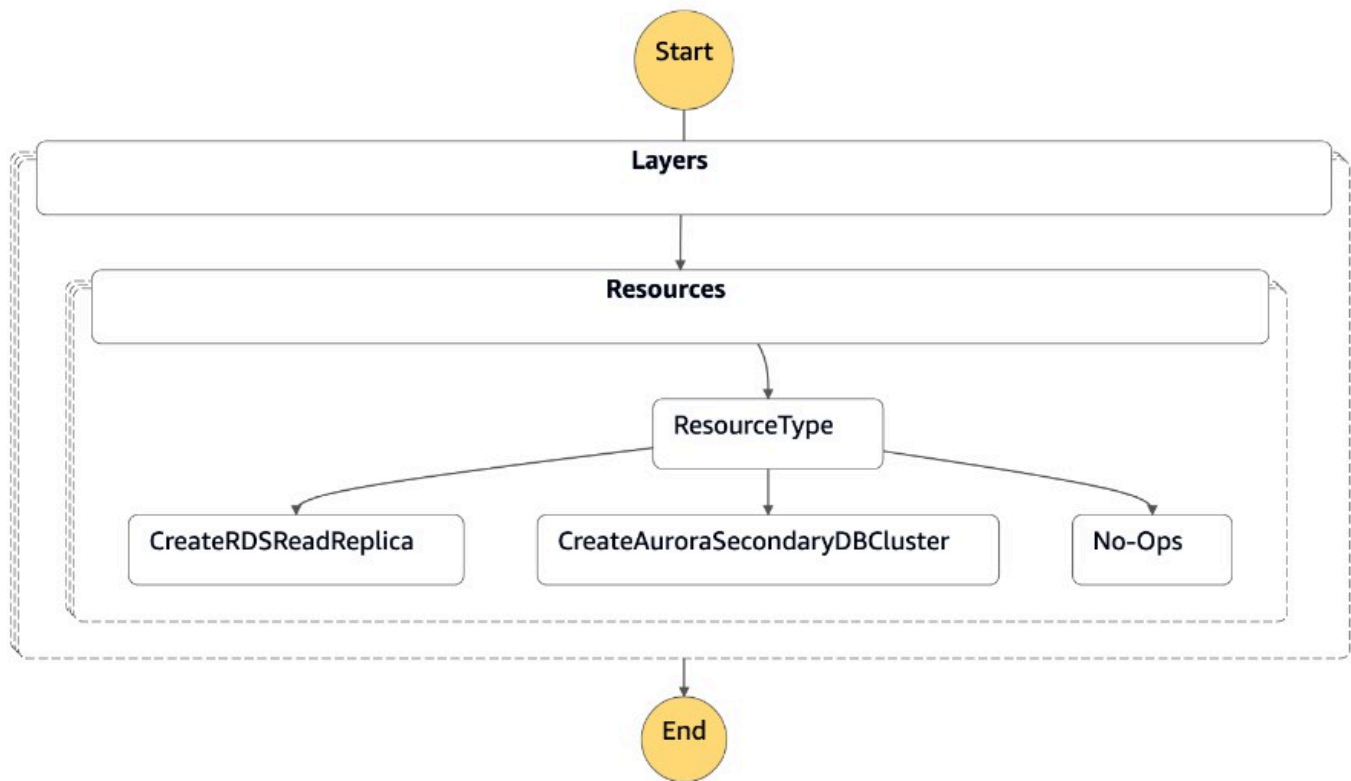
Parametername	Beschreibung	Erwartete Werte
layer(erforderlich: Zahl)	Die Reihenfolge der Verarbeitung. Alle in Layer 1 definierten Ressourcen müssen ausgeführt werden, bevor die Layer-2-Ressourcen ausgeführt werden.	1 oder 2 und so weiter

Parametername	Beschreibung	Erwartete Werte
Ressourcen (erforderlich: Wörterbuch-Array)	Alle Ressourcen innerhalb einer einzigen Schicht laufen parallel.	<pre>{   "resourceType": "String",   "resourceName": "String",   "parameters": {     "&lt;param1&gt;": "&lt;!Import cft-output-1&gt;",     ....   } }</pre>
resourceType (erforderlich: Zeichenfolge)	Typ der Ressource zur Identifizierung der Ressource	PromoteRDSReadReplica oder FailoverElastiCacheCluster
resourceName (optional: Zeichenfolge)	Um zu ermitteln, zu welchem Anwendungsportfolio diese Ressourcen gehören	Promote RDS for MySQL Read Replica
Parameter (erforderlich: Wörterbuch-Array)	Liste der Parameter, die für ein Failover oder ein Failback der AWS Datenbank erforderlich sind	<pre>{   "&lt;param1&gt;": "&lt;!Import cft-output-1&gt;",   "&lt;param2&gt;": "&lt;!Import cft-output-2&gt;",   ... }</pre>

## DR Orchestrator FAILBACK

Verwenden Sie die DR Orchestrator FAILBACK Zustandsmaschine nach dem DR-Ereignis, wenn die frühere primäre Region (us-east-1) aktiv ist. Sie können die [Read Replica](#) für Amazon RDS in der früheren primären Region aus der neuen primären Region (us-west-2) erstellen, um Ihrer DR-Strategie zu entsprechen. Da es sich um eine geplante Veranstaltung handelt, können Sie

diese Aktivität auf das Wochenende oder außerhalb der Hauptverkehrszeiten mit voraussichtlicher Ausfallzeit planen.



Die Zustandsmaschine akzeptiert Parameter in dem JSON Format, das im folgenden Code gezeigt wird:

```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "CreateRDSReadReplica",
          "resourceName": "Create RDS for MySQL Read Replica",
          "parameters": {
            "RDSInstanceIdentifier": "!Import rds-mysql-instance-identifier",
            "TargetClusterIdentifier": "!Import rds-mysql-instance-global-arn",
            "SourceRDSInstanceIdentifier": "!Import rds-mysql-instance-source-identifier",
            "SourceRegion": "!Import rds-mysql-instance-SourceRegion",
            "MultiAZ": "!Import rds-mysql-instance-MultiAZ",

```

```
"DBInstanceClass": "!Import rds-mysql-instance-DBInstanceClass",
"DBSubnetGroup": "!Import rds-mysql-instance-DBSubnetGroup",
"DBSecurityGroup": "!Import rds-mysql-instance-DBSecurityGroup",
"KmsKeyId": "!Import rds-mysql-instance-KmsKeyId",
"BackupRetentionPeriod": "7",
"MonitoringInterval": "60",
"StorageEncrypted": "True",
"EnableIAMDatabaseAuthentication": "True",
"DeletionProtection": "True",
"CopyTagsToSnapshot": "True",
"AutoMinorVersionUpgrade": "True",
"Port": "!Import rds-mysql-instance-DBPortNumber",
"MonitoringRoleArn": "!Import rds-mysql-instance-RDSMonitoringRole"
    }
  }
]
}
}
```

# Aurora-Zustände und Step Functions Functions-Zustandsmaschinen

In diesem Abschnitt werden die spezifischen Prozess- und Zustandsmaschinen für Failover und Failback von Amazon Aurora Aurora-Clustern behandelt. Die Cluster sind als globale Datenbank konfiguriert.

## Note

Zu Demonstrationszwecken wird in diesem Beispiel die Aurora MySQL-Compatible Edition verwendet. Sie können ähnliche Schritte für Aurora PostgreSQL-Compatible Edition verwenden.

## Stetiger Zustand

Im Steady-State wurde eine Amazon Aurora MySQL-kompatible globale Datenbank (`dr-globaldb-cluster-mysql`) mit zwei DB-Clustern erstellt. Der erste DB-Cluster (`db-cluster-01`) wurde in der Primärdatenbank (`us-east-1`) erstellt, um den AWS-Region Lese-/Schreib-Workload zu bedienen. Der zweite DB-Cluster (`db-cluster-02`) wurde in der sekundären Region (`us-west-2`) erstellt, um den schreibgeschützten Workload zu verarbeiten.

Zusätzlich zur Bereitstellung der DR-Lösung können Sie die Belastung Ihres primären DB-Clusters reduzieren, indem Sie Leseanfragen von Ihren Anwendungen an den sekundären DB-Cluster weiterleiten. Jeder dieser Cluster enthält eine Datenbank-Instance namens `dbcluster-01-use1-instance-1` und `dbcluster-02-usw2-instance-2`.

## Status des Ereignisses

Durch die Verwendung einer globalen Amazon Aurora Aurora-Datenbank können Sie Katastrophen relativ schnell planen und wiederherstellen. Die Wiederherstellung nach einem Notfall wird in der Regel anhand von Werten für Recovery Time Objective (RTO) und Recovery Point Objective (RPO) gemessen. Weitere Informationen finden Sie unter [Verwenden von Switchover oder Failover in einer globalen Amazon Aurora Aurora-Datenbank](#).

Bei einer globalen Aurora-Datenbank gibt es zwei verschiedene Ansätze für Failover:



- Switchover (verwaltetes geplantes Failover)
- Failover (manuelles, ungeplantes Failover oder Trennen und Heraufstufen)

## Umstellung

Switchover ist für kontrollierte Umgebungen vorgesehen, z. B. für betriebliche Wartungsarbeiten und andere geplante Betriebsabläufe. Mithilfe eines verwalteten geplanten Failovers können Sie den primären DB-Cluster Ihrer globalen Aurora-Datenbank in eine der sekundären Regionen verlagern. Da Switchover wartet, bis die sekundären DB-Cluster mit der Primärdatenbank synchronisiert sind, ist RPO 0 (kein Datenverlust). Weitere Informationen finden Sie unter [Durchführen von Switchovers für globale Amazon Aurora Aurora-Datenbanken](#).

Die `dr-orchestrator-stepfunction-FAILOVER` Zustandsmaschine wird während des Ereignisstatus aufgerufen, um Ihren primären Cluster in die von Ihnen gewählte sekundäre Region umzuschalten (`us-west-2`).

Gehen Sie wie folgt vor, um den Switchover durchzuführen:

1. Melden Sie sich bei der AWS Management Console an.
2. Ändern Sie die Region in die DR-Region (`us-west-2`).
3. Navigieren Sie zu Services und wählen Sie Step Functions aus.
4. Navigieren Sie zur `dr-orchestrator-stepfunction-FAILOVER` Zustandsmaschine.
5. Wählen Sie Ausführung starten und geben Sie den folgenden JSON-Code in den Input - optional Abschnitt ein:

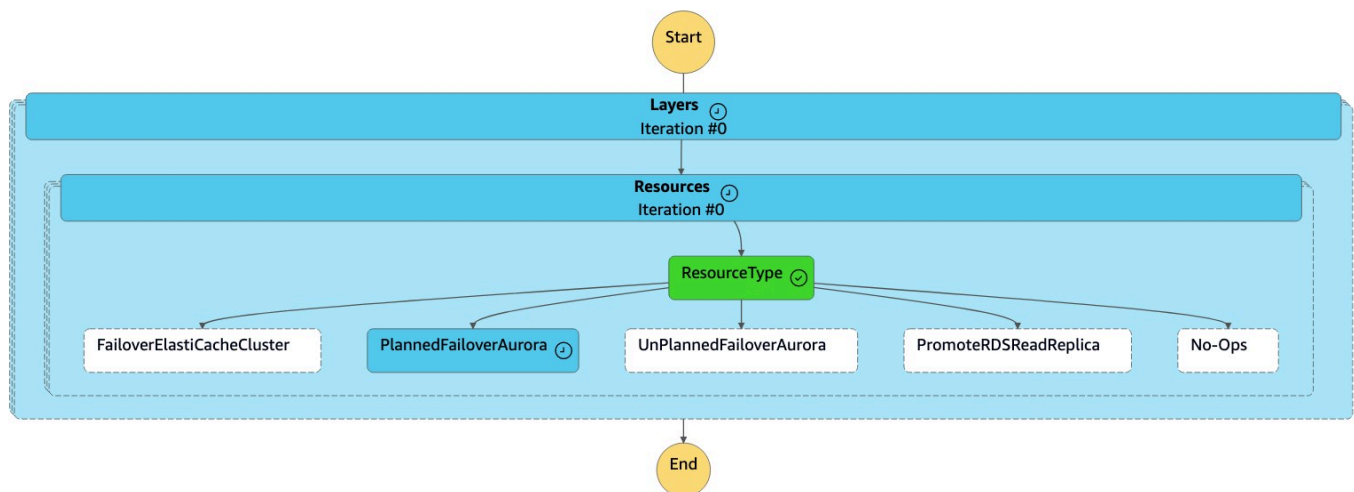
```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "PlannedFailoverAurora",
          "resourceName": "Switchover (planned failover) of Amazon Aurora global
databases (MySQL)",
          "parameters": {
            "GlobalClusterIdentifier": "!Import dr-globaldb-cluster-mysql-global-
identifier",
```

```

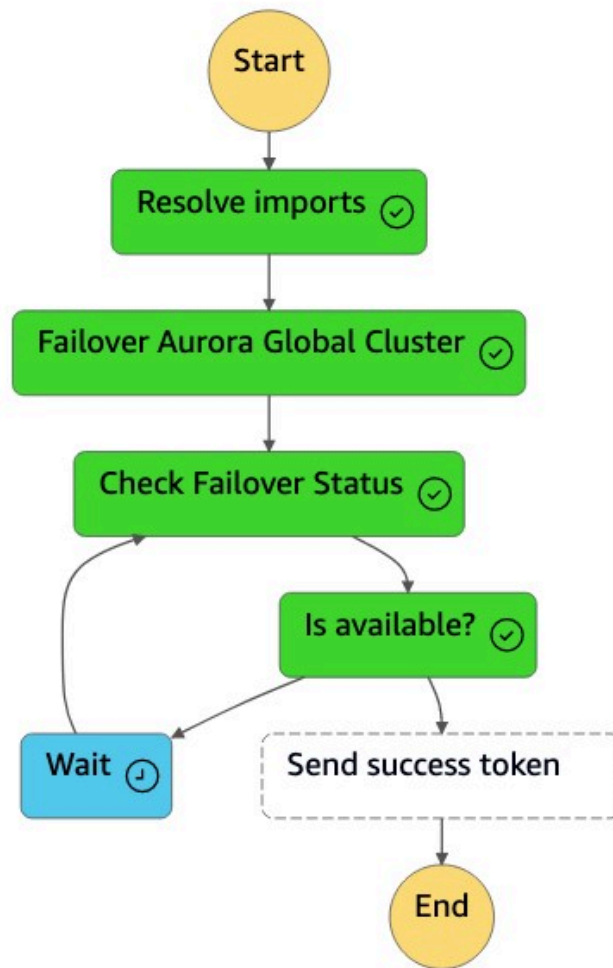
    "DBClusterIdentifier": "!Import dr-globaldb-cluster-mysql-cluster-
    identifier"
  }
}
]
}
]
}

```

6. Die `dr-orchestrator-stepfunction-FAILOVER` Zustandsmaschine liest den Ressourcentyp als `PlannedFailoverAuroraMySQL` und ruft die `dr-orchestrator-stepfunction-planned-Aurora-failover` Zustandsmaschine auf, um ein Failover für die globale Aurora-Datenbank durchzuführen.



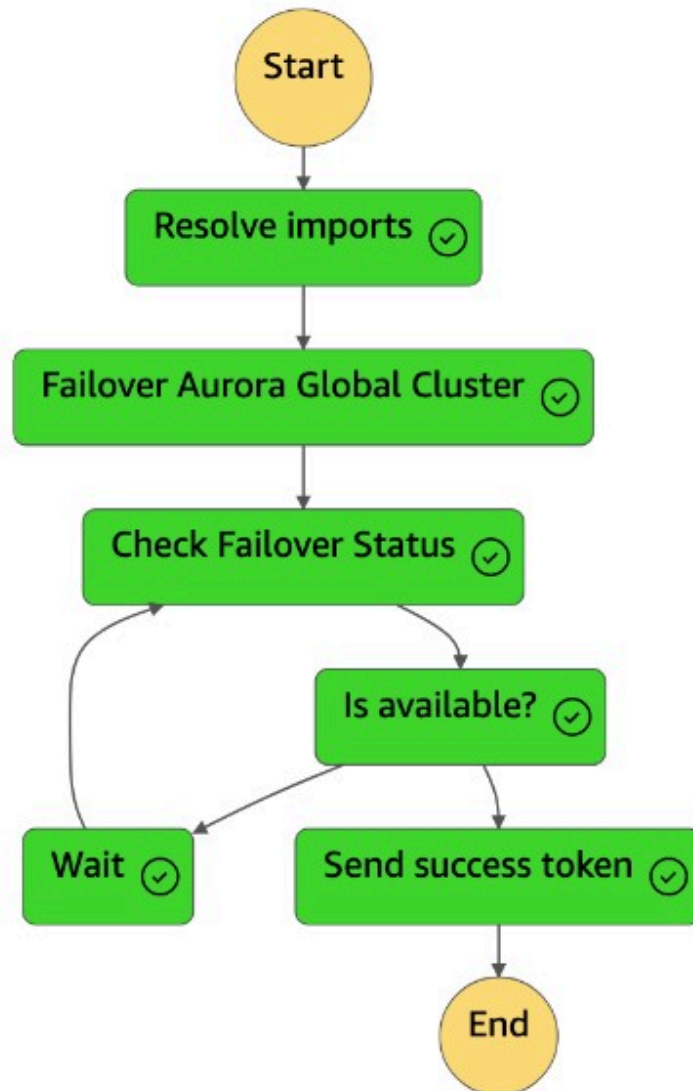
7. Die `dr-orchestrator-stepfunction-planned-Aurora-failover` State Machine führt die folgenden Schritte aus, um die Aurora MySQL-kompatible globale Datenbankrolle umzuschalten.



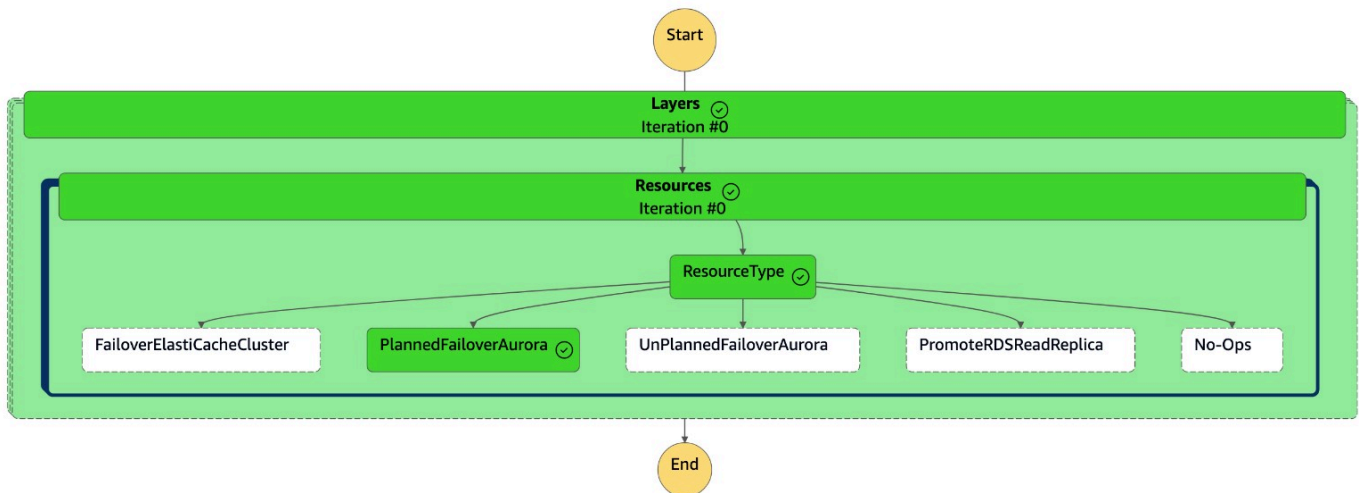
Schritt	Beschreibung	Erwartete Werte
Importe lösen	Eine Lambda-Funktion ersetzt !Import <variable name> Werte durch den tatsächlichen Namen.	<code>"!Import dr-global db-cluster-mysql-global-identifizier"</code> wird ersetzt durch <code>"dr-global db-cluster-mysql"</code> .

Schritt	Beschreibung	Erwartete Werte
Globaler Aurora-Failover-Cluster	Eine Lambda-Funktion ruft die Boto3-APIs <a href="#">failover_global_cluster</a> auf, um ein Failover für die globale Aurora-Datenbank durchzuführen.	<pre>{   'GlobalCluster': {     'GlobalClusterIdentifier': 'dr-global-cluster-mysql',     'GlobalClusterResourceId': 'cluster-cce7f9bec2846db4',     'GlobalClusterArn': 'arn:aws:rds::xxx',     'Status': 'failing-over',     ....   } }</pre>
Überprüfen Sie den Failover-Status	Eine Lambda-Funktion ruft <a href="#">describe_db_clusters</a> Boto3-APIs auf, um den Status des Failovers zu überprüfen.	modifizieren, verfügbar
Erfolgstoken senden	Eine Lambda-Funktion ruft <a href="#">send_task_success Boto3-APIs auf und sendet ein Erfolgstoken</a> zurück an die Zustandsmaschine. DR Orchestrator Failover	RiCdLtdH7x dMccoxlZF hglsdkzp /83P1E0 K9mBVKZSP7D9YRT1W

8. Navigieren Sie zur Amazon RDS-Konsole. Unter Status ändern sich die Werte für die globale Aurora-Datenbank von Verfügbar auf Umschalten oder Ändern.
9. Nachdem die `dr-orchestrator-stepfunction-planned-Aurora-failover` Zustandsmaschine abgeschlossen ist, sendet sie ein Erfolgstoken zurück an die `dr-orchestrator-stepfunction-FAILOVER` Zustandsmaschine.



10 Die dr-orchestrator-stepfunction-FAILOVER Zustandsmaschine ist abgeschlossen.



Auf der Konsole hat der sekundäre Cluster (`dbcluster-02`) jetzt die Rolle des primären Clusters, und der Cluster ist bereit, Lese-/Schreib-Workloads zu verarbeiten. Die Rolle des ursprünglichen primären Clusters (**`dbcluster-01`**) ist jetzt als Sekundärer Cluster aufgeführt.

## Manuelles ungeplantes Failover

In seltenen Fällen kann es bei Ihrer globalen Aurora-Datenbank zu einem unerwarteten Ausfall der AWS-Region Primärdatenbank kommen. In einem solchen Fall sind der primäre Aurora-DB-Cluster und sein Writer-Knoten nicht verfügbar, und die Replikation zwischen dem primären Cluster und den sekundären Clustern wird eingestellt. Um sowohl Ausfallzeiten (RTO) als auch Datenverlust (RPO) zu minimieren, sollten Sie schnell ein regionsübergreifendes Failover durchführen und Ihre globale Aurora-Datenbank rekonstruieren. Weitere Informationen finden Sie unter [Wiederherstellung einer globalen Amazon Aurora Datenbank nach einem ungeplanten Ausfall](#).

Um einen ungeplanten Failover durchzuführen, müssen Sie Ihren sekundären Cluster von der globalen Aurora-Datenbank trennen. Bevor Sie den ungeplanten Failover durchführen, beenden Sie Anwendungsschreibvorgänge auf Ihrem primären Aurora-DB-Cluster. Nachdem der Failover erfolgreich abgeschlossen wurde, konfigurieren Sie die Anwendung neu, sodass sie in den neuen primären DB-Cluster schreibt. Dieser Ansatz trägt dazu bei, Datenverlust zu verhindern. Es hilft auch, Dateninkonsistenzen zu vermeiden, wenn der primäre Writer-Knoten während des Failover-Prozesses wieder online geht.

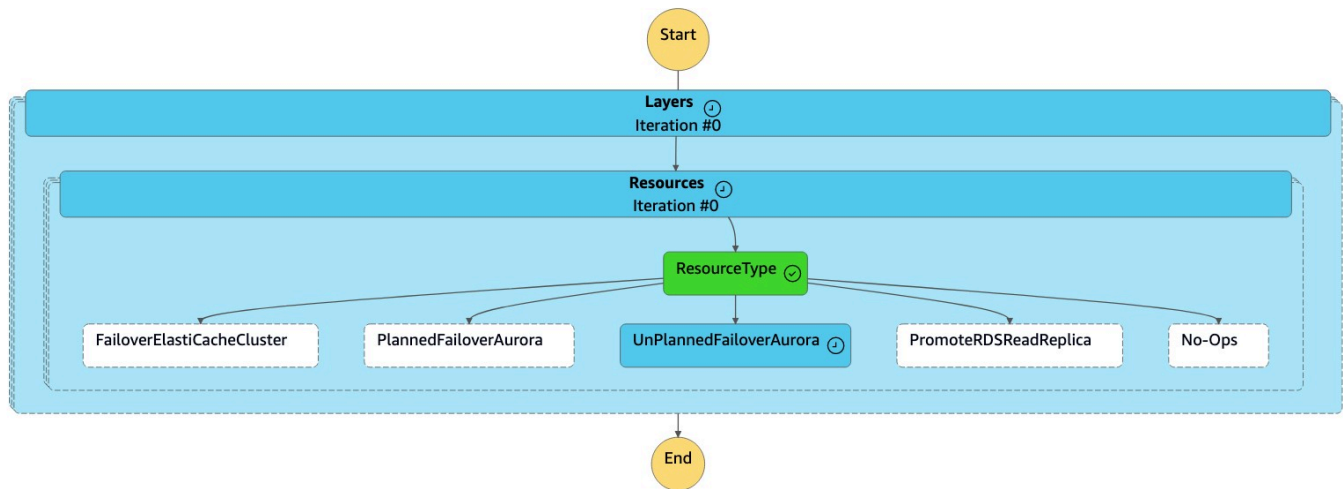
Rufen Sie den State Machine auf, um den ungeplanten Failover durchzuführen. `dr-orchestrator-stepfunction-FAILOVER` In diesem Beispiel befindet sich der sekundäre Cluster (`db-cluster-02`) in der DR-Region (`us-west-2`) im Steady-State.

Gehen Sie wie folgt vor, um den Failover durchzuführen:

1. Melden Sie sich in der -Konsole an.
2. Ändern Sie die Region in die DR-Region (us-west-2).
3. Navigieren Sie zu Services und wählen Sie Step Functions aus.
4. Navigieren Sie zur `dr-orchestrator-stepfunction-FAILOVER` Zustandsmaschine.
5. Wählen Sie Ausführung starten und geben Sie den folgenden JSON-Code in den Input - optional Abschnitt ein. Verwenden Sie `UnPlannedFailoverAurora` dabei `resourceType`:

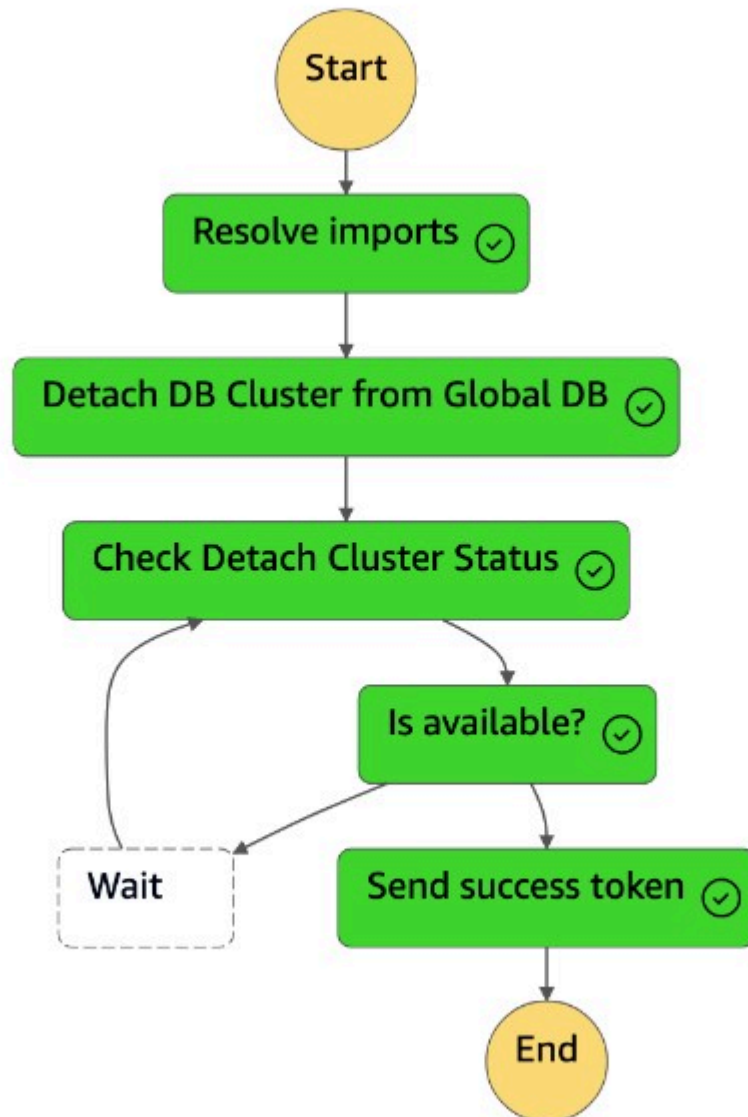
```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "UnPlannedFailoverAurora",
          "resourceName": "Performing unplanned failover for Amazon Aurora global
databases (MySQL)",
          "parameters": {
            "GlobalClusterIdentifier": "!Import dr-globaldb-cluster-mysql-global-
identifier",
            "DBClusterIdentifier": "!Import dr-globaldb-cluster-mysql-cluster-
identifier",
            "ClusterRegion": "!Import dr-globaldb-cluster-mysql-cluster-region"
          }
        }
      ]
    }
  ]
}
```

6. Die `dr-orchestrator-stepfunction-FAILOVER` Zustandsmaschine liest den Ressourcentyp als `UnPlannedFailoverAuroraMySQL` und ruft die Aufgabe `Detach Cluster from Global Database` von der `dr-orchestrator-stepfunction-unplanned-Aurora-failover` Zustandsmaschine aus auf.

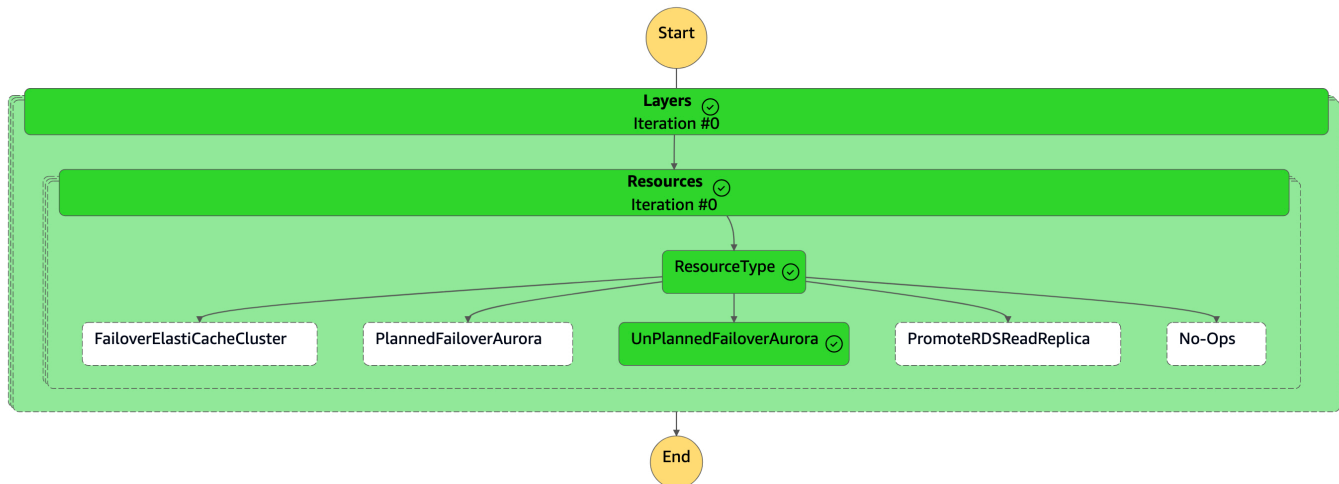


7. Die Detach Cluster from Global Database Aufgabe trennt (entfernt) den sekundären Cluster von der globalen Datenbank.





8. Der sekundäre Cluster (dbcluster-02) wird zu einem eigenständigen Cluster heraufgestuft und kann Lese-/Schreib-Workloads bedienen.
9. Die `dr-orchestrator-stepfunction-FAILOVER` Zustandsmaschine ist abgeschlossen.



10 Der sekundäre Cluster (`dbcluster-02`) ist von der globalen Aurora-Datenbank getrennt und wird zu einem eigenständigen Cluster, der die Lese-/Schreib-Arbeitslast bedient.

11 Konfigurieren Sie Ihre Anwendung neu, sodass alle Schreibvorgänge mithilfe des neuen Cluster-Endpunkts an diesen neuen eigenständigen Aurora-DB-Cluster gesendet werden.

## Failback

Ein Failback bringt Ihre Datenbank an den ursprünglichen (oder neuen) primären Standort zurück, nachdem ein Notfall (oder ein geplantes Ereignis) behoben wurde. Wenn der ungeplante Ausfall behoben ist, möchten Sie möglicherweise Ihre frühere Hauptregion wieder zur globalen Aurora-Datenbank hinzufügen. Sie müssen zuerst den vorhandenen DB-Cluster aus der früheren primären Region löschen, einen neuen DB-Cluster aus der neuen primären Region erstellen und dann den verwalteten geplanten Failover-Prozess verwenden, um die Rolle des neuen Clusters zu wechseln.

Dies kann als geplante Aktivität betrachtet werden, die Sie außerhalb der Hauptverkehrszeiten oder an einem Wochenende durchführen können.

Sie müssen [den Amazon Aurora Aurora-DB-Cluster manuell ändern](#) und deaktivieren, `DeletionProtection` bevor Sie den DR Orchestrator FAILBACK Zustandsmaschine aus der früheren primären Region (`us-east-1`) ausführen, da er mit `erstellt wurdeDeletionProtection`.

DR Orchestrator Framework verwendet die `dr-orchestrator-stepfunction-FAILBACK` Zustandsmaschine, um die Schritte zum Löschen des vorhandenen Clusters und zum Erstellen eines neuen Clusters in der ehemaligen Primärregion zu automatisieren.

Gehen Sie zum Deaktivieren DeletionProtection wie folgt vor:

1. Melden Sie sich in der -Konsole an.
2. Ändern Sie die Region in die frühere primäre Region (us-east-1).
3. Navigieren Sie zur Amazon RDS-Konsole, wählen Sie den Cluster-Namen (dbcluster-01) aus und klicken Sie auf Modify.
4. Deaktivieren Sie unter Löschschutz das Kontrollkästchen Löschschutz aktivieren und wählen Sie Weiter.
5. Wählen Sie Sofort anwenden und anschließend Cluster modifizieren aus.

Die DR Orchestrator FAILBACK Zustandsmaschine wird während des Failback-Vorgangs von der ehemaligen primären Region () us-east-1 aus aufgerufen.

Gehen Sie wie folgt vor, um das Failback durchzuführen:

1. Melden Sie sich in der -Konsole an.
2. Ändern Sie die Region in die frühere primäre Region (us-east-1).
3. Navigieren Sie zu Services und wählen Sie dann Step Functions aus.
4. Navigieren Sie zur DR Orchestrator FAILBACK Zustandsmaschine.
5. Wählen Sie Ausführung starten und geben Sie den folgenden JSON-Code in den Input - optional Abschnitt ein:

```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "CreateAuroraSecondaryDBCluster",
          "resourceName": "To create secondary Aurora MySQL Global Database Cluster",
          "parameters": {
            "GlobalClusterIdentifier": "!Import dr-globaldb-cluster-mysql-global-identifier",
            "DBClusterIdentifier": "!Import dr-globaldb-cluster-mysql-cluster-identifier",
            "DBClusterName": "!Import dr-globaldb-cluster-mysql-cluster-name",
            "SourceDBClusterIdentifier": "!Import dr-globaldb-cluster-mysql-source-cluster-identifier",

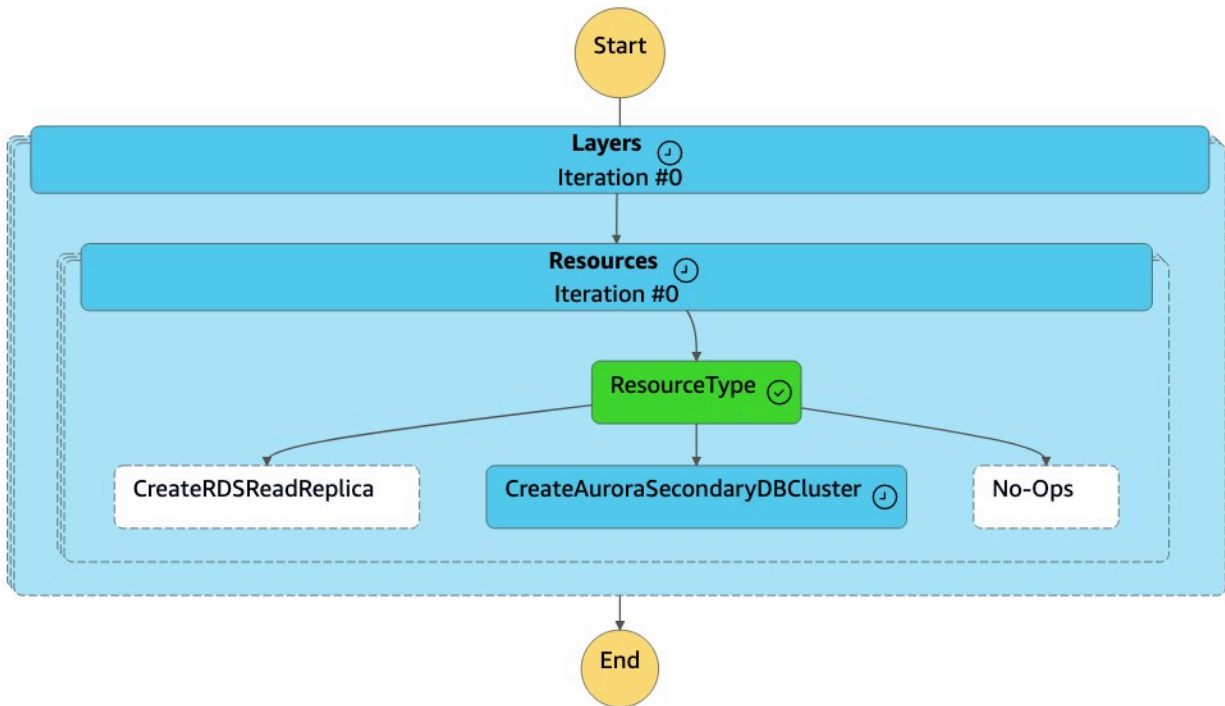
```

```

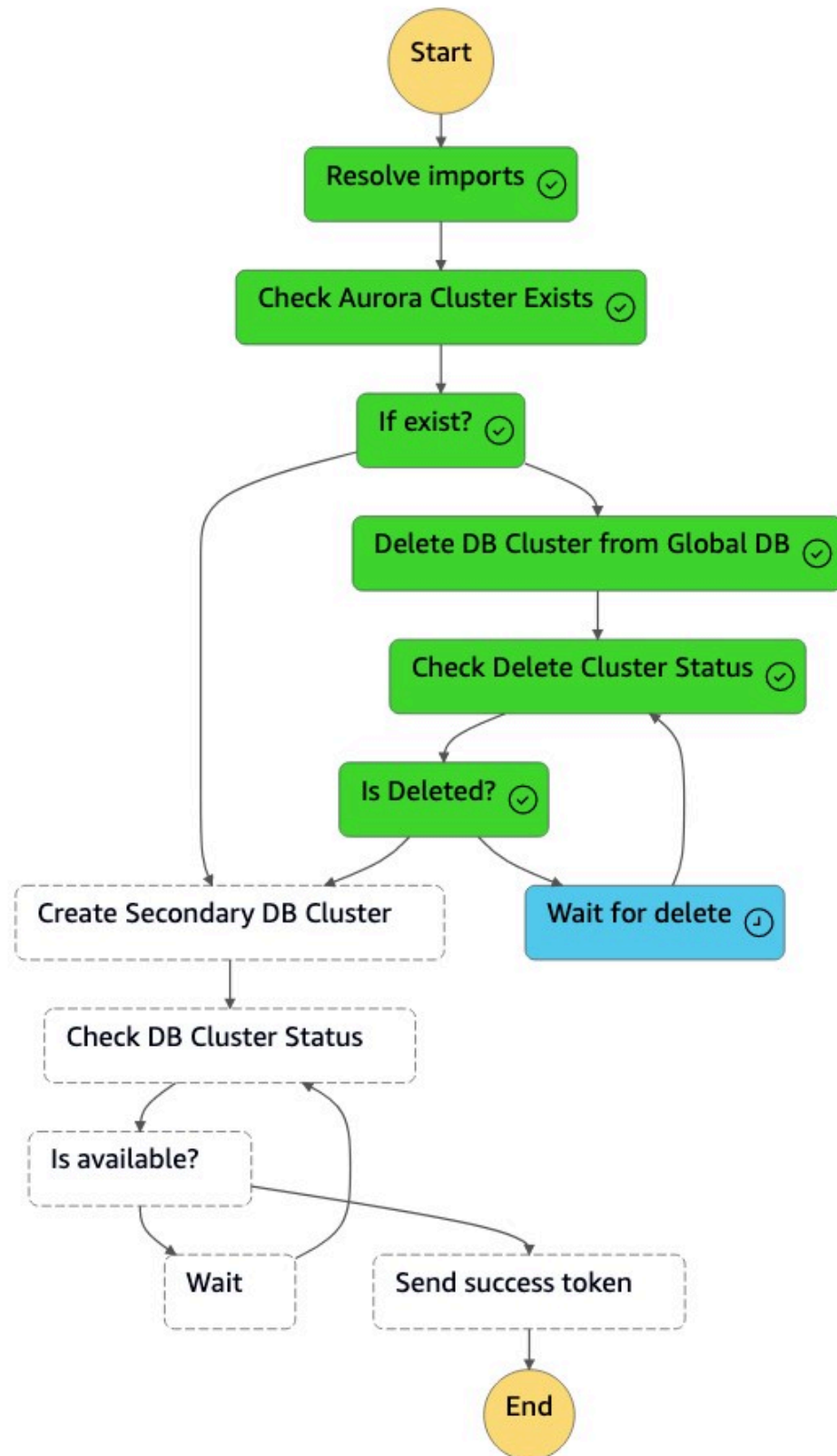
        "DBInstanceIdentifier": "!Import dr-globaldb-cluster-mysql-instance-
identifier",
        "Port": "!Import dr-globaldb-cluster-mysql-port",
        "DBInstanceClass": "!Import dr-globaldb-cluster-mysql-instance-class",
        "DBSubnetGroupName": "!Import dr-globaldb-cluster-mysql-subnet-group-
name",
        "VpcSecurityGroupIds": "!Import dr-globaldb-cluster-mysql-vpc-security-
group-ids",
        "Engine": "!Import dr-globaldb-cluster-mysql-engine",
        "EngineVersion": "!Import dr-globaldb-cluster-mysql-engine-version",
        "KmsKeyId": "!Import dr-globaldb-cluster-mysql-KmsKeyId",
        "SourceRegion": "!Import dr-globaldb-cluster-mysql-source-region",
        "ClusterRegion": "!Import dr-globaldb-cluster-mysql-cluster-region",
        "BackupRetentionPeriod": "7",
        "MonitoringInterval": "60",
        "StorageEncrypted": "True",
        "EnableIAMDatabaseAuthentication": "True",
        "DeletionProtection": "True",
        "CopyTagsToSnapshot": "True",
        "AutoMinorVersionUpgrade": "True",
        "MonitoringRoleArn": "!Import rds-mysql-instance-RDSMonitoringRole"
    }
}
]
}
]
}

```

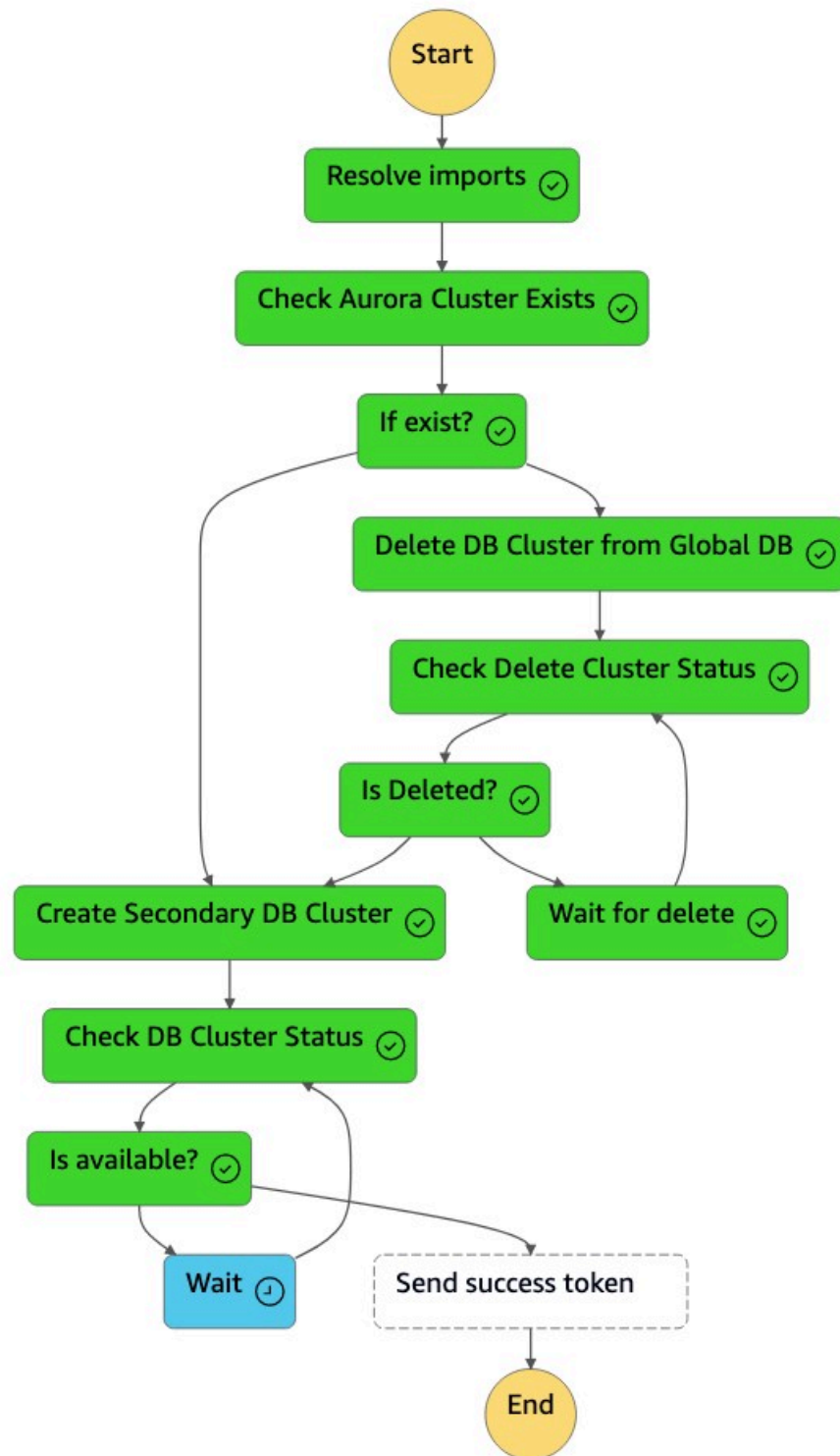
6. Die DR Orchestrator FAILBACK Zustandsmaschine liest den Ressourcentyp als `CreateAuroraSecondaryDBCluster` und ruft die `dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster` Zustandsmaschine auf.



7. Die `dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster` Zustandsmaschine löscht den vorhandenen Cluster (`dbcluster-01`) aus der früheren primären Region (`us-east-1`).

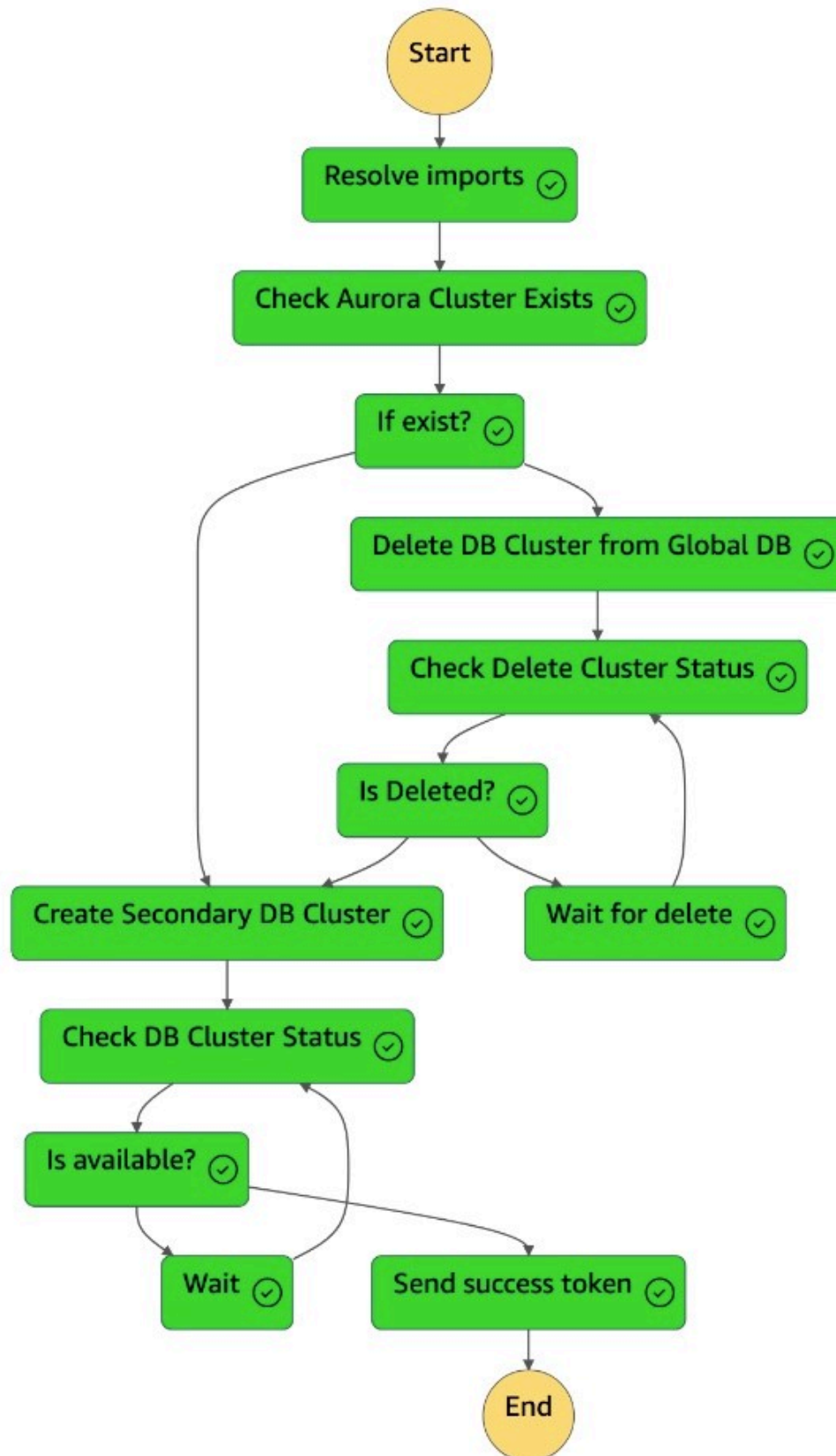


8. Nachdem der Cluster (dbcluster-01) gelöscht wurde, erstellt die Zustandsmaschine zusammen mit der DB-Instance einen neuen Cluster (dbcluster-01) und tritt der globalen Aurora-Datenbank als sekundärer Cluster bei, um schreibgeschützte Workloads zu bedienen.

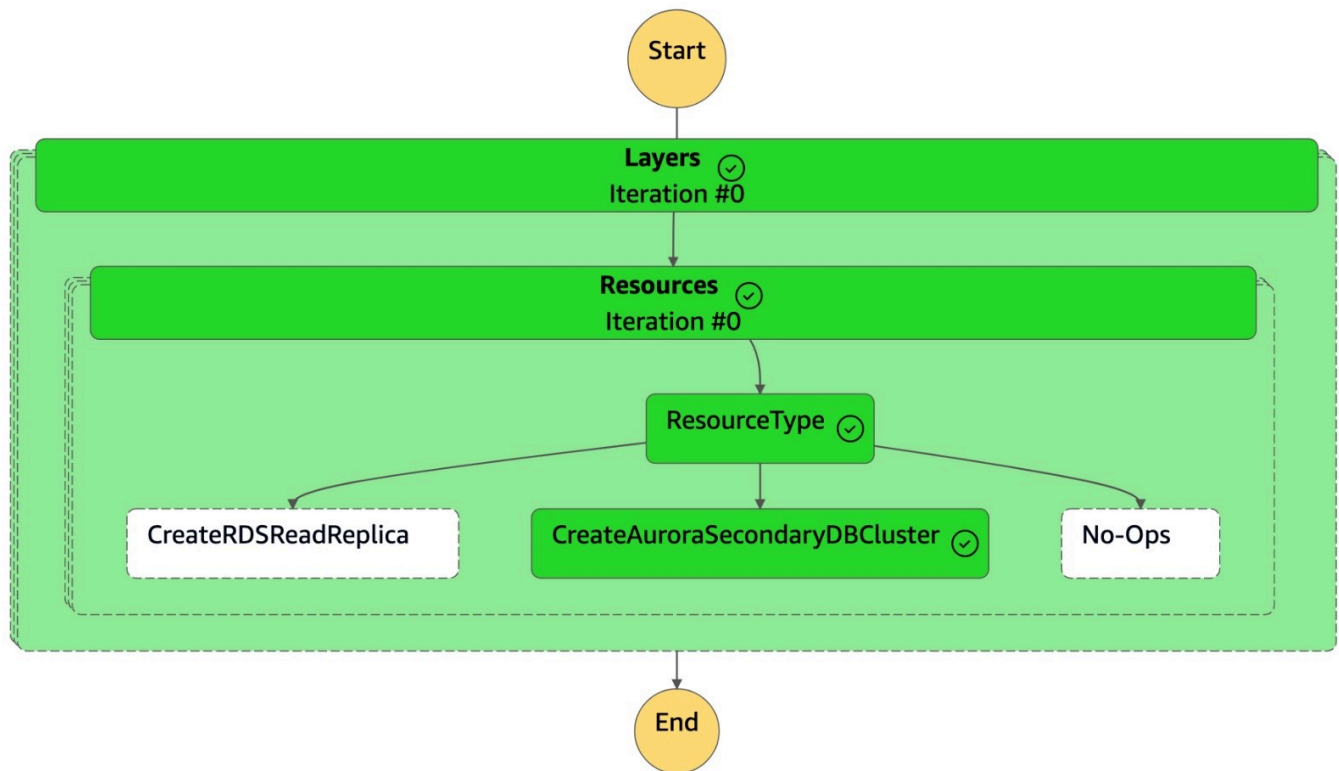


9. Sobald der sekundäre Cluster verfügbar ist, ist die `dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster` Zustandsmaschine fertig gestellt und sie sendet ein Erfolgstoken zurück an die DR Orchestrator Failback Zustandsmaschine.





10 Die dr-orchestrator-stepfunction-FAILBACK Zustandsmaschine ist abgeschlossen.



11 Sie können die globale Aurora-Datenbank auf der Amazon RDS-Konsole überprüfen.

Wenn Sie den primären DB-Cluster nach us-east-1 verlagern möchten, können Sie die im Abschnitt Switchover genannten Schritte ausführen.

# Stellen Sie die Lösung bereit

Nachdem Sie die Voraussetzungen eingerichtet haben, folgen Sie den Anweisungen im [DR Orchestrator Framework-Muster](#). Anweisungen für Amazon und Amazon RDS Aurora sind enthalten ElastiCache. Um die Lösung zu testen, überprüfen Sie die Voraussetzungen und Einschränkungen und stellen Sie Beispieldatenbanken auf bereit AWS.

## Voraussetzungen

- Aktiv [AWS-Konto](#) für die Bereitstellung dieser Lösung.
- Berechtigungen zum Erstellen von [Amazon Simple Storage Service \(Amazon S3\)](#) -Buckets und der in den [AWS CloudFormation](#) Vorlagen verwendeten Ressourcen.
- Bevor Sie die DR Orchestrator FAILBACK Zustandsmaschine ausführen:
  - [Ändern Sie die RDS Instance](#) manuell oder [ändern Sie den Amazon Aurora Aurora-DB-Cluster](#) und deaktivieren Sie ihn `DeletionProtection`.
  - Erstellen Sie vor dem Löschen der RDS Amazon-Instance oder des Aurora-Clusters einen [DB-Snapshot](#), wenn Sie eine Ursachenanalyse durchführen möchten (RCA).

## Einschränkungen

- Die Verfügbarkeit und der Support der Funktionen variieren je nach Version der einzelnen Datenbank-Engines und zwischen den einzelnen Versionen AWS-Regionen. Weitere Informationen zur Verfügbarkeit von Funktionen und Regionen für die regionsübergreifende Replikation finden Sie unter [Regionsübergreifende Read Replicas](#).
- Diese Lösung wurde mit einer primären Datenbankinstanz mit einer Read Replica getestet. Wenn Sie mehr als eine Read Replica verwenden möchten, testen Sie die Lösung gründlich, bevor Sie sie in einer Produktionsumgebung implementieren.
- Für Amazon RDS unterstützt die Lösung nur die folgenden Engines: [Amazon RDS for My SQL](#), [Amazon RDS for Postgre SQL](#) und [Amazon RDS for MariaDB](#).

## Stellen Sie Beispieldatenbanken bereit auf AWS

Um die regionsübergreifende DR-Lösung für Datenbanken zu demonstrieren AWS, können Sie [Amazon RDS for MySQL](#), [Amazon Aurora My SQL -Compatible Edition](#) und [Amazon ElastiCache \(RedisOSS\)](#) — Global Datastore bereitstellen. [Folgen Sie den Anweisungen in der App-](#)

[StackREADME Datei des -databases-Repositorys. aws-cross-region-dr](#) GitHub In diesem Beispiel ist us-east-1 die primäre Region und die sekundäre Region. us-west-2

Der Anwendungsstapel verwendet den Abschnitt AWS CloudFormation [Outputs](#), um die Parameter zu exportieren, die vom DR Orchestrator Framework zur Automatisierung von Failover und Failback von Datenbanken verwendet werden. AWS

# FAQ

Hier finden Sie Antworten auf Fragen zum DR Orchestrator Framework für Failover und Failback.

## Was sind die RPO und was kann RTO ich mit diesem Ansatz erreichen?

Informationen zu Recovery Time Objective (RTO) und Recovery Point Objective (RPO) finden Sie in der Strategie [Disaster Recovery Strategy for Databases unter AWS](#).

## Ist die Verwendung von AWS CloudFormation Exportvariablen obligatorisch?

Nein, Sie können den Wert der globalen Amazon Aurora Datenbank oder der Amazon RDS DB-Instance direkt im JSON Format übergeben (zum Beispiel:- "RDSInstanceIdentifier": "rds-mysql-instance").

## Kann ich den DR FAILOVER Orchestrator-Workflow verwenden, um ein Failover für mehr als eine AWS Datenbank durchzuführen?

Ja, Sie können mehr als eine Ressource in der Eingabeparameterdatei übergeben, um ein Failover für mehr als eine AWS Datenbank durchzuführen. Das folgende Codebeispiel zeigt ein paralleles Failover einer Amazon RDS for My SQL Read Replica und eines globalen Amazon ElastiCache (RedisOSS) -Datenspeichers:

```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "PromoteRDSReadReplica",
          "resourceName": "Promote RDS MySQL Read Replica",
          "parameters": {
            "RDSInstanceIdentifier": "!Import rds-mysql-instance-identifier",
```

```
        "TargetClusterIdentifier": "!Import rds-mysql-instance-global-arn"
      },
      {
        "resourceType": "FailoverElastiCacheCluster",
        "resourceName": "Failover ElastiCache Cluster",
        "parameters": {
          "GlobalReplicationGroupId": "!Import demo-redis-cluster-global-replication-
group-id",
          "TargetRegion": "!Import demo-redis-cluster-target-region",
          "TargetReplicationGroupId": "!Import demo-redis-cluster-target-replication-
group-id"
        }
      }
    ]
  }
}
```

## Wie kann ich den InvalidParameterCombination Fehler vermeiden, wenn ich die DR FAILBACK Orchestrator-Zustandsmaschine für Amazon RDS ausführe?

Der vollständige Text des Fehlers lautet:

```
"errorMessage": "An error occurred (InvalidParameterCombination) when
calling the DeleteDBInstance operation: Cannot delete protected DB
Instance, please disable deletion protection and try again."
```

Um den Fehler zu vermeiden, [ändern Sie die RDS Instanz](#), indem Sie DeletionProtection sie deaktivieren, bevor Sie die DR Orchestrator FAILBACK Zustandsmaschine ausführen.

## Nächste Schritte

Sie können DR Orchestrator Framework verwenden, um einen Genehmigungsprozess hinzuzufügen, bevor Sie die Failover- oder Failback-Aktivitäten einleiten. Wenn Sie beispielsweise den Failover während eines DR-Ereignisses einleiten, können Sie zusätzlich eine Zustandsmaschine hinzufügen, DR Orchestrator Failover um eine [Amazon Simple Notification Service \(Amazon SNS\) - Benachrichtigung](#) per E-Mail zu senden. Sobald die Genehmigung erteilt wurde, beginnt die Failover-Aktivität.

Das Anwendungs-Failover kann auch in das DR Orchestrator Framework integriert werden, um den DNS-Eintrag umzuschalten und Anwendungen auf den neuen Endpunkt der Datenbankinstanz oder des Clusters zu verweisen.

Sie können die [AWS X-Ray](#) Ablaufverfolgung verwenden, um die Failover-Dauer für die Berechnung des RTO zu ermitteln. Sie können ein Monitoring-Dashboard auf X-Ray aufbauen.

# Ressourcen

- [Replikation AWS-Regionen über globale Datenspeicher hinweg](#) (Amazon ElastiCache (RedisOSS) -Dokumentation)
- [Verwenden von Failover in einer Amazon Aurora Global Database](#) (Aurora-Dokumentation)
- [Automatisieren Sie das Amazon Aurora Global Database-Endpunktmanagement für geplante und ungeplante Failover](#) (Blogbeitrag)
- [Notfallwiederherstellung: 3 Failover-Szenarien für Ihre globale Amazon Aurora Aurora-Datenbank mit Terraform \(Teil 2\)](#) (Blogbeitrag)
- [Notfallwiederherstellungsstrategie für Datenbanken auf AWS](#) (AWS Prescriptive Guidance Strategy)
- [Automatisieren Sie regionsübergreifendes Failover und Failback mithilfe des DR Orchestrator Framework](#) (Prescriptive Guidance Pattern)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="#">Erste Veröffentlichung</a>	—	3. Mai 2024

# AWS Glossar zu Prescriptive Guidance

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

## Zahlen

### 7 Rs

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

- **Faktorwechsel/Architekturwechsel** – Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile cloudnativer Feature nutzen, um Agilität, Leistung und Skalierbarkeit zu verbessern. Dies beinhaltet in der Regel die Portierung des Betriebssystems und der Datenbank. Beispiel: Migrieren Sie Ihre On-Premises-Oracle-Datenbank zu der SQL Postgre-kompatible Amazon-Aurora-Edition.
- **Plattformwechsel (Lift and Reshape)** – Verschieben Sie eine Anwendung in die Cloud und führen Sie ein gewisses Maß an Optimierung ein, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre On-Premises-Oracle-Datenbank zu Amazon Relational Database Service (AmazonRDS) für Oracle im 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 Customer Relationship Management (CRM) zu Salesforce.com.
- **Hostwechsel (Lift and Shift)** – Verschieben Sie eine Anwendung in die Cloud, ohne Änderungen vorzunehmen, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre On-Premises-Oracle-Datenbank zu Oracle auf einer EC2 Instance im 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 von 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.

### abstrakte Dienste

Siehe [Managed Services](#).

### ACID

Siehe [Atomizität, Konsistenz, Isolierung, Haltbarkeit](#).

### Aktiv-Aktiv-Migration

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

### Aktiv-Passiv-Migration

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

### Aggregationsfunktion

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

## AI

Siehe [künstliche Intelligenz](#).

## AIOps

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

## Anonymisierung

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

## Anti-Muster

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

## Anwendungssteuerung

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

## Anwendungsportfolio

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

## künstliche Intelligenz (KI)

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

## Operationen mit künstlicher Intelligenz (AIOps)

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

## Asymmetrische Verschlüsselung

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

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

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

## Attributbasierte Zugriffskontrolle () ABAC

Die Praxis, detaillierte Berechtigungen auf der Grundlage von Benutzerattributen wie Abteilung, Aufgabenrolle und Teamname zu erstellen. Weitere Informationen finden Sie unter [ABAC für 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 eindeutiger Standort in einer AWS-Region, der isoliert und somit vor Ausfällen in anderen Availability Zones geschützt ist und kostengünstige Netzwerkkonnektivität mit geringer Latenz zu anderen Availability Zones in der gleichen Region bereitstellt.

## AWS Cloud Adoption Framework (AWS CAF)

Ein Rahmen von Leitlinien und bewährten Verfahren von AWS, um Organisationen bei der Entwicklung eines effizienten und effektiven Plans für eine erfolgreiche Umstellung auf die Cloud zu unterstützen. AWS CAF unterteilt die Beratung in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Geschäft, Menschen, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Für diese Perspektive AWS CAF bietet es Beratung für Personalentwicklung, Training und Kommunikation, um das Unternehmen

auf eine erfolgreiche Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie [AWS CAF auf der Website](#) und im [AWS CAF Whitepaper](#).

## AWS Workload Qualification Framework (AWS WQF)

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

## B

### Bad 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 Anrufe 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.

## Bot-Netz

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

## branch

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

## Zugang durch Glasbruch

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

## Brownfield-Strategie

Die bestehende Infrastruktur in Ihrer Umgebung. Wenn Sie eine Brownfield-Strategie für eine Systemarchitektur anwenden, richten Sie sich bei der Gestaltung der Architektur nach den

Einschränkungen der aktuellen Systeme und Infrastruktur. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und [Greenfield](#)-Strategien mischen.

## Puffer-Cache

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

## Geschäftsfähigkeit

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

## Planung der Geschäftskontinuität (BCP)

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

# C

## CAF

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

## Canary-Deployment

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 unter [Cloud-Kompetenzzentrum](#).

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

## CI/CD

Weitere Informationen finden Sie unter [Kontinuierliche Bereitstellung und kontinuierliche Integration](#).

## Klassifizierung

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

## clientseitige Verschlüsselung

Lokale Verschlüsselung von Daten bevor der Ziel- 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 hier: [CCoEBeiträge](#) 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 üblicherweise verbunden mit [Edge-Computing-Technologie](#).

## 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 Organisationen normalerweise durchlaufen, wenn sie auf AWS Cloud die

- 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 einer CCoE, Einrichtung eines Betriebsmodells)
- Migration – Migrieren einzelner Anwendungen
- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

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

## CMDB

Siehe [Datenbank für das Konfigurationsmanagement](#).

## Code-Repository

Ein Ort, an dem Quellcode und andere Komponenten wie Dokumentation, Beispiele und Skripts gespeichert und im Rahmen von Versionskontrollprozessen aktualisiert werden. Zu den gängigen Cloud-Repositorys gehören GitHub oder 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. Die Verlagerung dieser Daten in leistungsschwächere und kostengünstigere Speicherstufen oder -klassen kann die Kosten senken.

## 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.

## Konfigurations-Drift

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 Phase der Migration, die sich CMDB in der Phase der Portfolioerkennung und -analyse befindet.

## Konformitätspaket

Eine Sammlung von AWS Config -Regeln und Abhilfemaßnahmen, die Sie zusammenstellen können, um Ihre Compliance- und Sicherheitsprüfungen individuell anzupassen. Mit Hilfe einer YAML Vorlage können Sie ein Konformitätspaket als einzelne Einheit in einer AWS-Konto und Region oder in einer gesamten Organisation bereitstellen. Weitere Informationen finden Sie unter [Konformitätspakete](#) in der AWS Config -Dokumentation.

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

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

## Daten-Drift

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

## Daten während der Übertragung

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

## data Mesh

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.

## Datensubjekt

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 in anwenden AWS, fügen Sie mehrere Steuerelemente auf verschiedenen Ebenen der AWS Organizations -Struktur hinzu, um das Backup von Ressourcen zu unterstützen. Ein defense-in-depth Ansatz könnte beispielsweise Multi-Faktor-Authentifizierung, Netzwerksegmentierung und Verschlüsselung kombinieren.

## delegierter Administrator

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

## Bereitstellung

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

## Entwicklungsumgebung

Siehe [Umgebung](#).

## Detektivische Kontrolle

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

## 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.

## Dimensionstabelle

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

## Katastrophe

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

## Disaster Recovery (DR)

Die Strategie und der Prozess, mit denen Sie Ausfallzeiten und Datenverluste aufgrund einer [Katastrophe](#) minimieren. Weitere Informationen finden Sie unter [Notfallwiederherstellung von Workloads unter Notfallwiederherstellung AWS von Workloads unter Notfallwiederherstellung](#) von AWS Workected 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 [Modernisieren älterer Microsoft ASP NET\(ASMX\) schrittweise Webservices mithilfe von Containern und Amazon API Gateway](#).

## DR

Siehe [Disaster Recovery](#).

### Erkennung von Abweichungen

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

## DVSM

Siehe [Abbildung der Wertströme in der Entwicklung](#).

## E

### EDA

Siehe [explorative Datenanalyse](#).

### 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) Prinzipalen erstellen AWS PrivateLink und diesen Berechtigungen gewähren. Diese Konten oder Prinzipale können eine private Verbindung zu Ihrem Endpunkt-Service herstellen, indem sie VPC Schnittstellen-Endpunkte erstellen. Weitere Informationen finden Sie unter [Einen Endpunkt-Service erstellen](#) in der Amazon Virtual Private Cloud (AmazonVPC) -Dokumentation.

## Unternehmensressourcenplanung (ERP)

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

## Envelope-Verschlüsselung

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

## Umgebung

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

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

- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

## Epics

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

## ERP

Weitere Informationen finden Sie unter [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

### Fakten-Tabelle

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.

### scheitern

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

### Fehlerisolationsgrenze

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 [AWS Fehler-Isolationsgrenzen](#).

## 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 Machine Learning mit:AWS](#).

## Featuretransformation

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

## FGAC

Siehe [differenzierte Zugriffskontrolle](#).

## differenzierte Zugriffskontrolle () FGAC

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

## Flash-Cut-Migration

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

# G

## Geo-Blockierung

Siehe [geografische Einschränkungen](#).

## Geografische Einschränkungen (Geoblocking)

In Amazon eine Option CloudFront, die verhindert, 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 unter [Einschränken der geografischen Verteilung von Inhalt](#) in der CloudFront -Dokumentation.

## Gitflow-Workflow

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

## Greenfield-Strategie

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

## Integritätsschutz

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

# H

## HEKTAR

Siehe [Hochverfügbarkeit](#).

### Heterogene Datenbankmigration

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

### hohe Verfügbarkeit (HA)

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

### Modernisierung des Historikers

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 for 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 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 unter [Einsatz einer unveränderlichen Infrastruktur](#) in AWS Well-Architected Framework.

## Eingehende (ingress) VPC

In einer AWS -Multi-Konto-Architektur, VPC die Netzwerkverbindungen von außerhalb einer Anwendung akzeptiert, überprüft und weiterleitet. Die [Referenzarchitektur für die AWS Sicherheit](#) empfiehlt, Ihr Netzwerkkonto mit eingehenden und ausgehenden Nachrichten und Inspektionssäule einzurichten, VPCs 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. Weitere Informationen finden Sie unter [Aufbau einer digitalen Transformationsstrategie für das industrielle Internet der Dinge \(IIoT\)](#).

## Inspektion VPC

In einer AWS -Multi-Konto-Architektur, eine zentralisierte Architektur, VPC die Inspektionen des Netzwerkverkehrs zwischen VPCs (in demselben oder unterschiedlichen AWS-Regionen), das Internet und On-Premises-Netzwerke verwaltet. Die [-Referenzarchitektur für die AWS Sicherheit](#) empfiehlt, Ihr Netzwerkkonto mit eingehenden und ausgehenden Nachrichten und Inspektionssäule einzurichten, VPCs 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-Informationsbibliothek (ITIL)

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

## IT-Servicemanagement (ITSM)

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

## ITIL

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

## ITSM

Siehe [IT-Servicemanagement](#).



# L

## Etikettenbasierte Zugriffskontrolle ( ) LBAC

Eine Implementierung der obligatorischen Zugriffskontrolle (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, skalierbare und sichere AWS -Umgebung mit mehreren Konten. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und 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.

## LBAC

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 -Dokumentation](#). IAM

## 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.

## Managed Services

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.

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

## MES

Siehe [Manufacturing Execution System](#).

## Message Queuing Telemetry Transport (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 klar definiert kommuniziert APIs und in der Regel kleinen, eigenständigen Teams gehört. Ein Versicherungssystem kann beispielsweise Microservices beinhalten, die Geschäftsfunktionen wie Vertrieb oder Marketing oder Subdomains wie Einkauf, Schadenersatz oder Analytik zugeordnet sind. Zu den Vorteilen von Microservices gehören Agilität, flexible Skalierung, einfache Bereitstellung, wiederverwendbarer Code und Ausfallsicherheit. Weitere Informationen finden Sie unter [Integrieren von Microservices mithilfe von AWS -Serverless-Services](#).

## Microservices-Architekturen

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

## Migration Acceleration Program (MAP)

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

## Migration in großem Maßstab

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

### Migrationsfabrik

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

### Migrationsmetadaten

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

### Migrationsmuster

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

### Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration zu AWS Cloud MPAbietet eine detaillierte Portfoliobewertung (richtige Servergröße, TCO Preisgestaltung, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung, Migrationspriorisierung und Wellenplanung). Das [MPATool](#) (erfordert Anmeldung) ist für alle AWS -Berater und APN Partnerberater kostenlos verfügbar.

## Migration Readiness Assessment (MRA)

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

## Migrationsstrategie

Der Ansatz, der verwendet wird, um einen Workload auf den zu migrieren AWS Cloud. Weitere Informationen finden Sie unter [7 Rs](#) in diesem Glossar und [Ihre Organisation mobilisieren, um groß angelegte 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 im AWS Cloud](#).

## Bewertung der Modernisierungsfähigkeit

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

## 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](#).

### OCM

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 Betriebsebene](#).

## Online-Migration

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

## OPC-UA

Siehe [Offene Prozesskommunikation — Einheitliche Architektur](#).

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

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

## Vereinbarung auf Betriebsebene () OLA

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

## 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

Eine Spur, AWS CloudTrail die von erstellt wird und alle Ereignisse für alle AWS-Konten in einer Organisation in protokolliert 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 unter [Erstellen eines Pfads für eine Organisation](#) in der CloudTrail -Dokumentation.

## Organisatorisches Veränderungsmanagement (OCM)

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

## Ursprungszugriffskontrolle (OAC)

In CloudFront, eine erweiterte Option für die Einschränkung des Zugriffs zur Sicherung Ihrer Amazon Simple Storage Service (Amazon S3) -Inhalte. OACunterstützt 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 Einschränkung des Zugriffs zur Sicherung Ihrer Amazon-S3-Inhalte. CloudFront Erstellt einen PrinzipalOAI, mit dem sich Amazon S3 authentifizieren kann, wenn Sie verwenden. Authentifizierte Prinzipale können nur über eine bestimmte CloudFront Distribution auf Inhalte in einem S3-Bucket zugreifen. Siehe auch [OAC](#), das bietet eine detailliertere und verbesserte Zugriffskontrolle.

## ORR

Siehe [Überprüfung der Betriebsbereitschaft](#).

## NICHT

Siehe [Betriebstechnologie](#).



## Ausgehend (egress) VPC

In einer AWS -Multi-Konto-Architektur, VPC die Netzwerkverbindungen verarbeitet, die von einer Anwendung aus initiiert werden. Die [Referenzarchitektur für die AWS Sicherheit](#) empfiehlt, Ihr Netzwerkkonto mit eingehenden und ausgehenden Nachrichten und Inspektionssäule einzurichten, VPCs 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 Entitys](#) in der IAM -Dokumentation.

### Personenbezogene Daten (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 hierfür PII sind Namen, Adressen und Kontaktinformationen.

### PII

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 kann (siehe [identitätsbasierte Richtlinie](#)), geben Sie die Zugriffsbedingungen an (siehe [ressourcenbasierte Richtlinie](#)), oder definieren Sie die maximalen Berechtigungen für alle Konten in einer Organisation in AWS Organizations (siehe Richtlinie zur [Servicekontrolle](#)).

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

Weitere Informationen finden Sie in der IAM Dokumentation unter Principal in [Roles \(Begriffe und Konzepte\)](#).

## 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 Domains reagieren soll. Weitere Informationen finden Sie unter [Arbeiten mit privat gehosteten Zonen](#) in der Route-53-Dokumentation.

## proaktive Steuerung

Eine [Sicherheitskontrolle](#), die den Einsatz von nicht konformen Ressourcen verhindern soll. Diese Steuerelemente scannen Ressourcen, 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, von der Konstruktion, Entwicklung und Markteinführung über Wachstum und Reife bis hin zu Verkauf und Verkauf.

## Produktionsumgebung

Siehe [Umgebung](#).

## programmierbare Logiksteuerung (PLC)

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 Microservice-basierten System kann ein Microservice beispielsweise Ereignismeldungen in einem Kanal veröffentlichen [MES](#), 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 SQL relationalen 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

### RACImatrix

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.

### RASCImatrix

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.

## Architekturwechsel

Siehe [7 Rs.](#)

## Recovery Point Objective (RPO)

Die maximal zulässige Zeitspanne seit dem letzten Datenwiederherstellungspunkt. Damit wird festgelegt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Serviceunterbrechung gilt.

## Ziel für die Wiederherstellungszeit (RTO)

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

## Faktorwechsel

Siehe [7 Rs.](#)

## Region

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

## Hostwechsel

Siehe [7 Rs.](#)

## Veröffentlichung

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

umziehen

Siehe [7 Rs.](#)

Wechsel der 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 Ausfallsicherheit](#).

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.

Matrix (verantwortlich, rechenschaftspflichtig, konsultiert, informiert RACI)

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 RASCIMatrix bezeichnet, und wenn Sie sie ausschließen, wird sie als RACIMatrix 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 Prozess, bei dem Sie das [Secret](#) in regelmäßigen Abständen aktualisieren. So erschweren Sie einem Angreifer den Zugriff auf die Anmeldeinformationen.

Zugriffskontrolle für Zeilen und Spalten (RCAC)

Die Verwendung einfacher, flexibler SQL Ausdrücke, die über definierte Zugriffsregeln verfügen. 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

SAML2.0

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

SCADA

Siehe [Aufsichtskontrolle und Datenerfassung](#).

## SCP

Siehe [Richtlinie zur Dienstkontrolle](#).

## Secret

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

## 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 (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 EC2 Amazon-Instance oder das Rotieren von Anmeldeinformationen.

## Serverseitige Verschlüsselung

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



## Service-Kontrollrichtlinie (SCP)

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

## Service-Endpunkt

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

## Service Level Agreement () SLA

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

## Indikator für das Serviceniveau () SLI

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

## Ziel auf Serviceniveau () 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 [System zur Verwaltung von Sicherheitsinformationen und Ereignissen](#).

## zentraler Fehlerpunkt (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](#).

## SLO

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

## split-and-seed Modell

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

## SPOF

Siehe [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 [Modernisieren älterer MicrosoftASP.NET\(ASMX\) schrittweise Webservices mithilfe von Containern und Amazon API Gateway](#).

## Subnetz

Ein Bereich von IP-Adressen in Ihrem 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.

## Synthetische Tests

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

# T

## tags

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

## Zielvariable

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

## Aufgabenliste

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

## Testumgebungen

[Siehe Umgebung.](#)

## Training

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

## Transit-Gateway

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

## Stammbasierter Workflow

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

## Vertrauenswürdiger Zugriff

Erteilen von Berechtigungen für einen Service, den Sie für die Ausführung von Aufgaben in AWS Organizations und in ihren Konten und in Ihrem Namen in Ihrer Organisation angeben. Der vertrauenswürdige Service erstellt in jedem Konto eine mit dem Service verknüpfte Rolle, wenn diese Rolle benötigt wird, um Verwaltungsaufgaben für Sie auszuführen. Weitere Informationen finden Sie 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.

## VPCPeering

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 VPC Amazon-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.

## Fensterfunktionen

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.

## WORM

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

## WQF

Siehe [AWSWorkload-Qualifizierungsrahmen](#).

einmal schreiben, viele lesen (WORM)

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

## Z

### Zero-Day Exploit

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

### Zero-Day-Sicherheitslücke

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

### Zombie-Anwendung

Eine Anwendung, deren durchschnittliche CPU 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.