



自动执行关系数据库灾难恢复解决方案 AWS

AWS 规范性指导



AWS 规范性指导：自动执行关系数据库灾难恢复解决方案 AWS

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

简介	1
目标	1
概述	2
故障转移过程	2
#####	2
故障恢复过程	4
DR Orchestrator ##恢复架构	4
灾难恢复协调器故障转移	5
参数详细信息	6
灾难恢复协调器故障恢复	7
Aurora 状态机和 Step Functions 状态机	10
稳定状态	10
事件状态	10
切换	10
手动计划外故障转移	16
故障恢复	19
部署解决方案	27
常见问题解答	28
使用这种方法我能实现的 RPO 和 RTO 是什么？	28
必须使用 AWS CloudFormation 导出变量吗？	28
我能否使用 DR Orchestrator 故障转移工作流程对多个 AWS 数据库进行故障切换？	28
当我运行适用于 Amazon RDS 的 DR Orchestrator 故障恢复状态机时，如何避免出现 InvalidParameterCombination 错误？	29
后续步骤	30
资源	31
文档历史记录	32
.....	xxxiii

自动执行关系数据库的灾难恢复解决方案 AWS

亚马逊 Web Services 的 Jitendra Kumar、Oliver Francis 和 Pavithra Balasubramanian ()AWS

2024 年 5 月 ([文档历史记录](#))

在 Amazon Web [Services \(\)](#) 上[规划灾难恢复 \(DR AWS\) 策略](#)时，您可以为组织中的数据库实施灾难恢复 (DR) 解决方案。您可以通过配置事件驱动的架构来执行数据库实例的故障转移，从而实现灾难恢复解决方案的自动化。这种方法的好处包括以下几点：

- 减少因手动协调而导致的人为错误
- 在某些情况下提供尽可能短的恢复时间目标 (RTO)
- 促进灾难恢复解决方案的可重复测试

如果没有自动化，则主服务器 AWS 区域中断将需要组织中的资源手动执行运行手册中的步骤。这些步骤可能包括多个流程，例如在辅助数据库中提升 Amazon Relational Database Service (Amazon RDS) 只读副本 AWS 区域 以用作新的主数据库实例。AWS 提供[AWS 适用于 Python 的 SDK \(Boto3\)](#) APIs 来自动执行这些操作，这些操作可以构成满足组织需求的灾难恢复解决方案的组成部分。

本指南讨论了一种以清单为导向的集中式灾难恢复协调器，它可以帮助您在同一个 AWS 数据库中 AWS 区域 实现灾难恢复解决方案。AWS 账户执行故障转移的决定仍需要人工干预，并且超出了自动化的范围。

本指南的目标受众是应用程序架构师、基础设施架构师、数据库架构师和数据库顾问。

目标

实施 DR Orchestrator 框架解决方案可以帮助您实现以下成果：

- 减少停机期间的手动错误 — 灾难恢复过程的手动执行很容易出错。在灾难恢复活动的各个阶段自动执行操作可以减少手动错误，并有助于确保更顺畅的故障转移或故障恢复。
- 大规模管理灾难恢复 — 如果您的组织拥有大量数据库，那么构建灾难恢复自动化可以帮助您扩展灾难恢复解决方案。您可以使用 DR Orchestrator 框架自动执行 AWS 数据库灾难恢复解决方案。DR Orchestrator 使用 JSON 格式的参数文件自动执行 AWS 数据库的故障转移或故障恢复步骤。

灾难恢复协调器框架概述

DR Orchestrator Framework 提供一键式解决方案，用于协调和自动执行数据库的跨区域灾难恢复。AWS 它使用[AWS Step Functions](#)和[AWS Lambda](#)来执行故障切换和故障恢复期间所需的步骤。Step Functions 状态机为协调器设计中的决策提供了基础。用于执行故障转移或故障恢复操作的 API 操作被编码到从状态机内部调用的 Lambda 函数中。Lambda 函数运行 [适用于 Python \(Boto3\) 的 AWS SDK](#) API 来与数据库进行交互。AWS

DR Orchestrator Framework 包含两台主状态机，分别对应于故障转移和故障恢复阶段。

对于 Amazon RDS，故障转移阶段将跨区域 RDS 只读副本提升为独立数据库实例。对于 Amazon Aurora 来说，当主区域在罕见的意外中断期间出现故障时，其写入节点不可用。写入器节点和辅助群集之间的复制将停止。您必须将辅助群集与全局数据库分离，并将其提升为独立群集。应用程序可以连接并向独立群集发送写入流量。您可以使用相同的过程将全球数据库的[主数据库集群切换](#)到辅助区域。对于受控场景，请使用此方法，例如：

- 运营维护
- 计划中的操作程序
- 将亚马逊 ElastiCache (Redis OSS) 辅助集群提升为新的主集群

故障恢复阶段在活动主区域和新的辅助区域之间建立数据的实时复制。

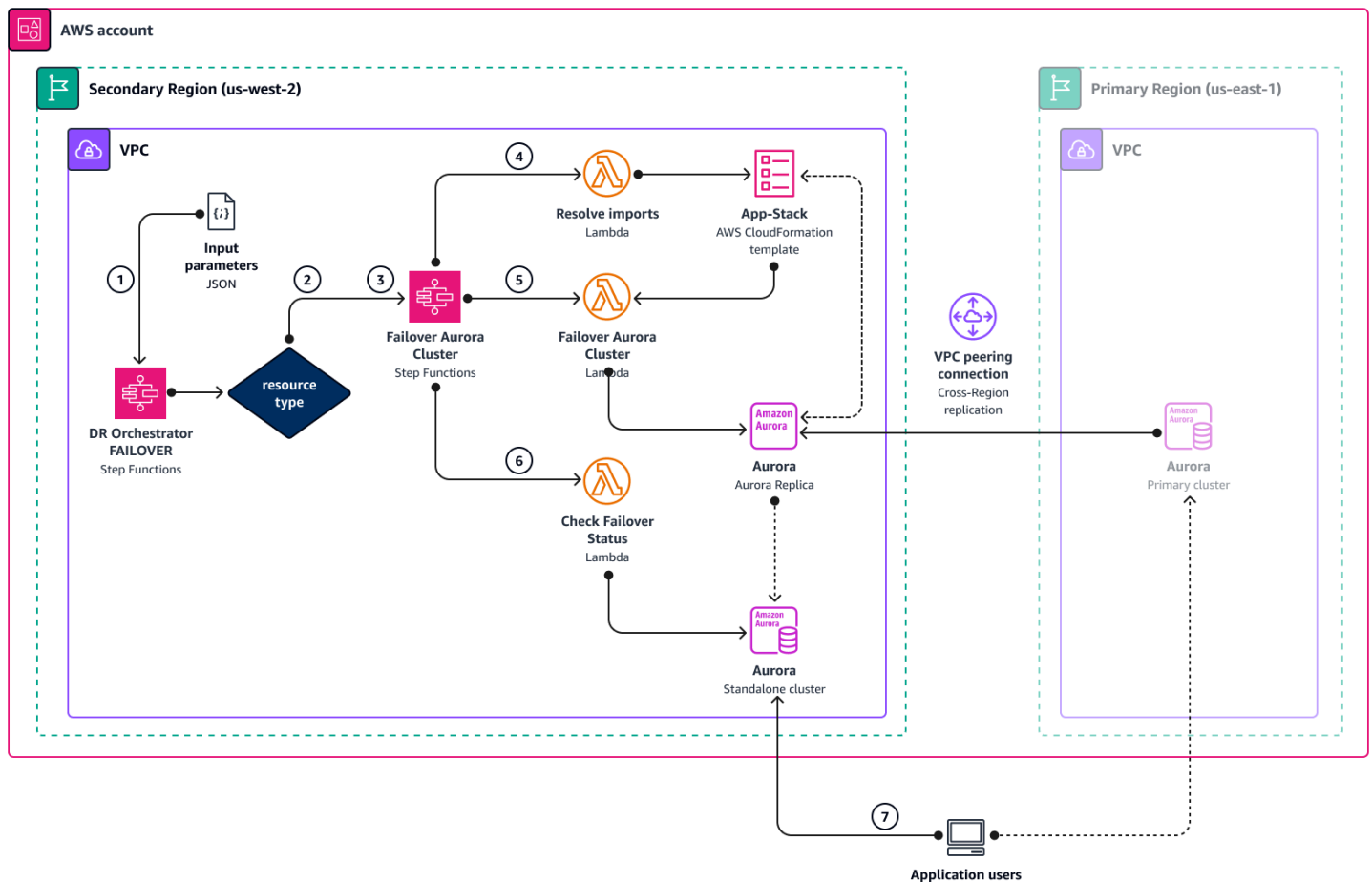
重要的是要明白 DR Orchestrator 仅适用于数据库。所有引用这些数据库且位于同一区域的应用程序可能需要单独的串联故障转移解决方案。在数据库故障转移到辅助区域后，需要更新应用程序以连接到将用作数据源的新数据库实例。

故障转移过程

要执行故障转移，请运行DR Orchestrator FAILOVER状态机。在此阶段，辅助区域中已经存在辅助数据库，可以作为只读副本 (Amazon RDS)，也可以作为辅助集群 (Amazon Aurora)。当您运行DR Orchestrator FAILOVER状态机时，它会将辅助数据库提升为主数据库。

#####

下图显示了使用 DR Orchestrator 时 Amazon Aurora 的故障转移过程的概念。Amazon Aurora 和亚马逊 ElastiCache 使用相同的工作流程，但使用不同的状态机和 Lambda 函数。



1. DR Orchestrator FAILOVER状态机读取输入的 JSON 参数。
2. 根据resourceType参数，状态机调用其他状态机：Promote RDS Read ReplicaFailover Aurora Cluster、或Failover ElastiCache。如果在输入中传递了多个资源，则这些状态机将并行运行。
3. Failover Aurora Cluster状态机在以下三个步骤中分别调用 Lambda 函数。
4. L Resolve imports ambda 函数使用模板中的实际值"! import <export-variable-name>"进行解析。App-Stack AWS CloudFormation
5. **Failover Aurora Cluster**Lambda 函数将只读副本提升为独立数据库实例。
6. L **Check Failover Status**ambda 函数检查已推广的数据库实例的状态。状态变为“可用”后，Lambda 函数将成功令牌发送回调用状态机并完成。
7. 您可以将应用程序重定向到灾难恢复区域 (us-west-2) 中的独立数据库，该数据库现在是主数据库。

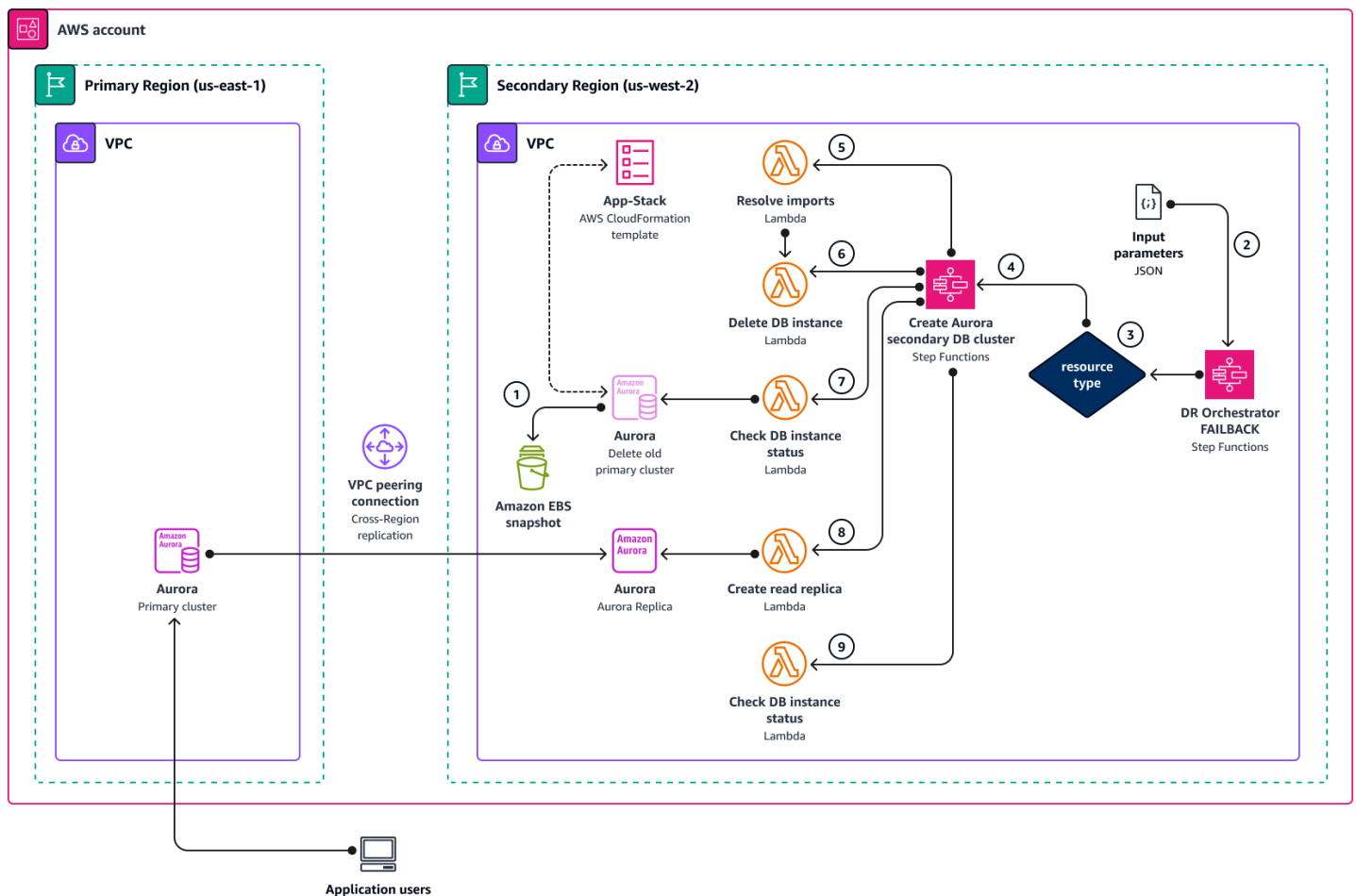
故障恢复过程

在您以前的主区域 (us-east-1) 再次启动后，您可以回切到该区域，以便中的数据库再次us-east-1成为主区域。要启动故障恢复，请运行DR Orchestrator FAILBACK状态机。顾名思义，此状态机开始将您的新主区域 (us-west-2) 中的更改复制回以前的主区域 (us-east-1)，后者充当当前的辅助区域。

在两个区域之间建立复制后，您可以启动故障恢复。要回切并返回到您的原始主区域 (us-east-1)，请在当前辅助区域 (us-east-1) 中运行DR Orchestrator FAILOVER状态机以将其升级到主区域。

DR Orchestrator ##恢复架构

下图显示了使用 DR Orchestrator 时 Amazon Aurora 的故障恢复过程的概念。



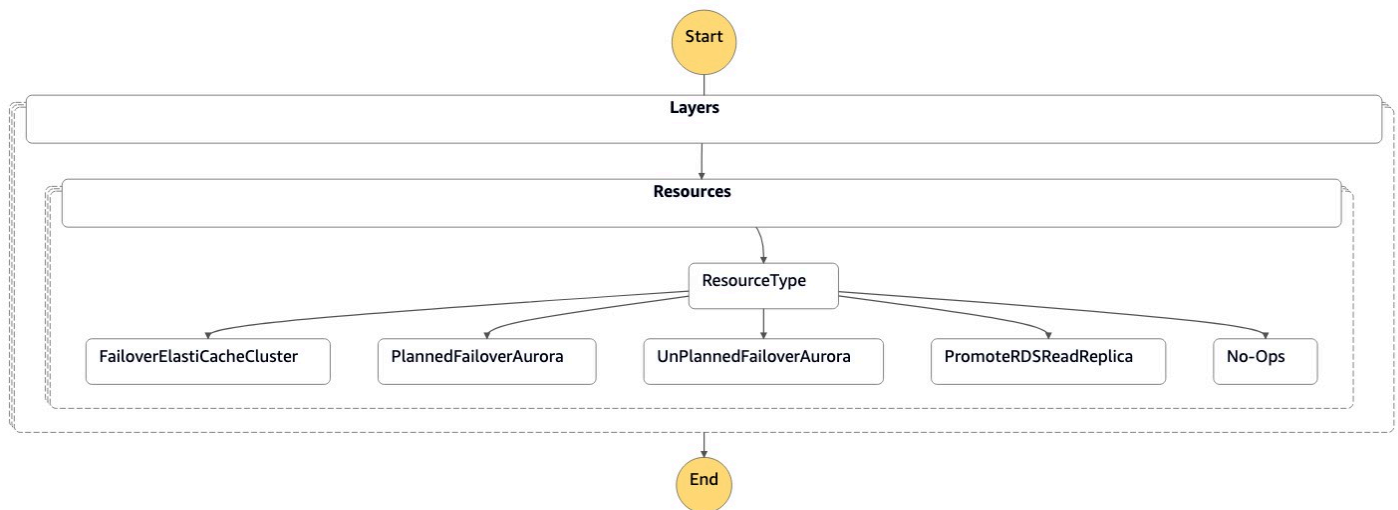
1. 在开始故障恢复之前，请手动拍摄数据库快照，以便在执行根本原因分析 (RCA) 时使用。

- 此外，在之前的主区域 (us-east-1) 中对 Aurora 集群禁用 DeletionProtection。
2. DR Orchestrator FAILBACK状态机读取输入的 JSON 参数。
 3. 基于此resourceType，DR Orchestrator FAILBACK状态机调用Create Aurora Secondary DB Cluster状态机。
 4. Create Aurora Secondary DB Cluster状态机通过以下五个步骤中的每一个步骤调用 Lambda 函数。
 5. L Resolve import ambda 函数使用模板中的实际值"! import <export-variable-name>"进行解析。App-Stack CloudFormation
 6. L Delete DB Instance ambda 函数会删除以前的主实例。
 7. Check DB instance statusLambda 函数会进行检查，Delete DB Instance status直到数据库被删除。
 8. Create Read ReplicaLambda 函数从新的主区域中的数据库实例在辅助区域中创建只读副本。
 9. Check DB instance statusLambda 函数检查只读副本数据库实例的状态。当状态为“可用”时，Lambda 函数将成功令牌发送回调调用状态机，调用状态机已完成。

灾难恢复协调器故障转移

在主区域 (us-east-1) 关闭时或计划内事件（例如操作维护）期间，在灾难恢复事件中使用DR Orchestrator FAILOVER状态机。

可以调用该函数并行地对单个或多个数据库进行故障切换。



状态机接受 JSON 格式的参数，如以下代码所示：

```
{
  "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"
          }
        }
      ]
    }
  ]
}
```

参数详细信息

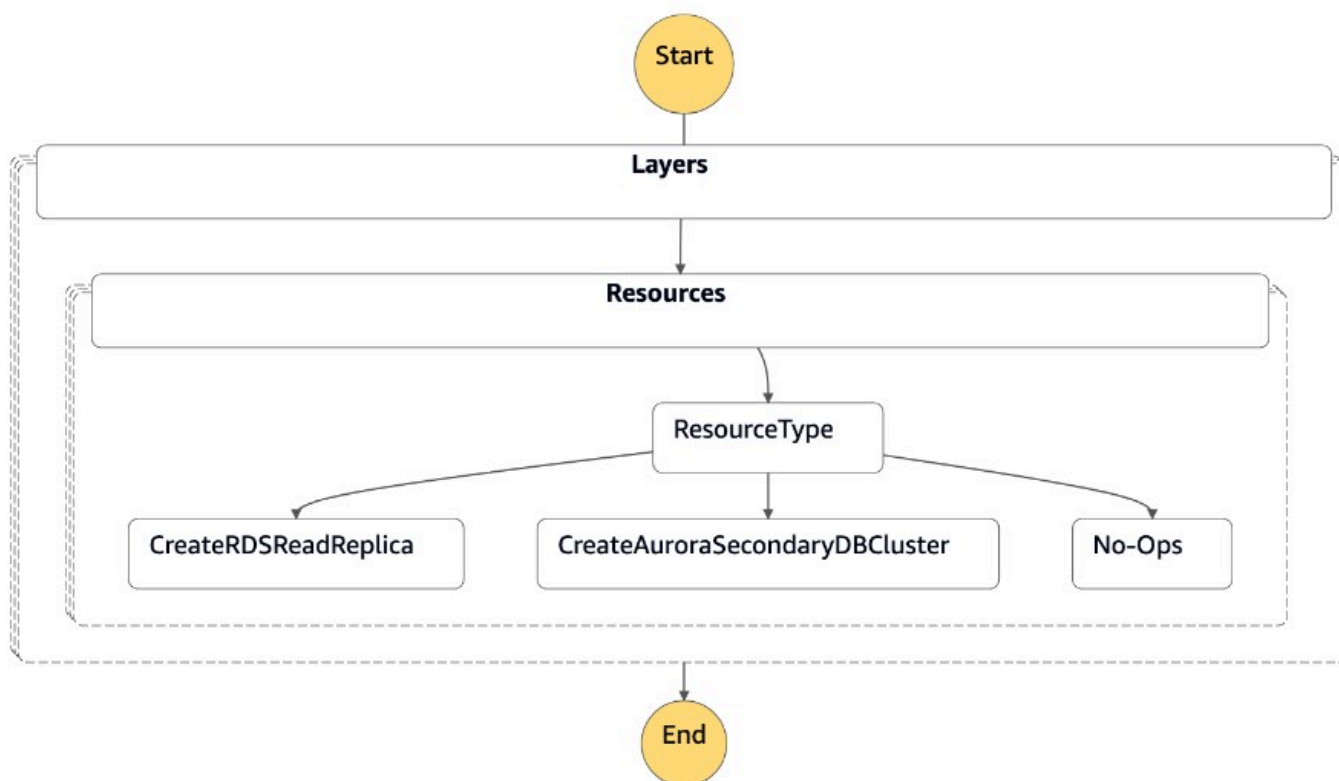
下表显示了DR Orchestrator FAILOVER状态机使用的参数。

参数名称	说明	预期值
layer (必填 : 数字)	处理顺序。第 1 层中定义的所有资源都必须先运行，然后才能运行第 2 层资源。	1 或 2，依此类推
资源 (必填 : 字典数组)	单层中的所有资源并行运行。	{

参数名称	说明	预期值
		<pre>"resourceType": "String", "resourceName": "String", "parameters": { "<param1>": "<!Import cft-output-1>", }</pre>
resourceType （必填：字符串）	用于标识资源的资源类型	PromoteRDSReadReplica 或 FailoverElastiCacheCluster
resourceName （可选：字符串）	确定这些资源属于哪个应用程序组合	Promote RDS for MySQL Read Replica
参数（必填：字典数组）	对 AWS 数据库进行故障切换或故障恢复所需的参数列表	<pre>{ "<param1>": "<!Import cft-output-1>", "<param2>": "<!Import cft-output-2>", }</pre>

灾难恢复协调器故障恢复

在灾难恢复事件之后使用DR Orchestrator FAILBACK状态机，即以前的主区域(us-east-1) 已启动。您可以从新的主区域(us-west-2) 为以前的主区域的 Amazon RDS 创建[只读副本](#)，以符合您的灾难恢复策略。由于这是计划内活动，因此您可以将此活动安排在周末或非高峰工作时间进行，预计会有停机时间。



状态机接受 JSON 格式的参数，如以下代码所示：

```

{
  "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 状态机和 Step Functions 状态机

本节介绍特定于故障转移和故障恢复 Amazon Aurora 集群的过程和状态机。集群被配置为全局数据库。

Note

出于演示目的，此示例使用了 Aurora MySQL-Compatible 版。您可以对 Aurora PostgreSQL-Compatible 版使用类似的步骤。

稳定状态

在稳定状态下，已创建了一个包含两个数据库集群的 Amazon Aurora MySQL-Compatible 全球数据库 (dr-globaldb-cluster-mysql)。已在主集群 (db-cluster-01) 中创建了第一个数据库集群 AWS 区域 (us-east-1) 来 read/write 处理工作负载。第二个数据库集群 (db-cluster-02) 已在辅助区域 (us-west-2) 中创建，用于处理只读工作负载。

除了提供灾难恢复解决方案外，您还可以通过将应用程序中的读取查询路由到辅助数据库集群来减轻主数据库集群的负载。每个集群都分别包含一个名为 dbcluster-01-use1-instance-1 和 dbcluster-02-usw2-instance-2 的数据库实例。

事件状态

通过使用 Amazon Aurora 全球数据库，您可以相当快地规划灾难并从灾难中恢复。灾难恢复通常使用恢复时间目标 (RTO) 和恢复点目标 (RPO) 的值来衡量。有关更多信息，请参阅在 [Amazon Aurora 全球数据库中使用切换或故障转移](#)。

使用 Aurora 全局数据库，有两种不同的故障转移方法：

- 切换 (托管计划内故障转移)
- 故障转移 (手动计划外故障切换，或分离并升级)

切换

切换适用于受控环境，例如操作维护和其他计划中的操作程序。通过使用托管计划内故障转移，您可以将 Aurora 全球数据库的主数据库集群重新定位到一个辅助区域。由于切换会等到辅助数据库集群与主

数据库同步，因此 RPO 为 0 (无数据丢失)。要了解更多信息，请参阅对 [Amazon Aurora 全球数据库执行切换](#)。

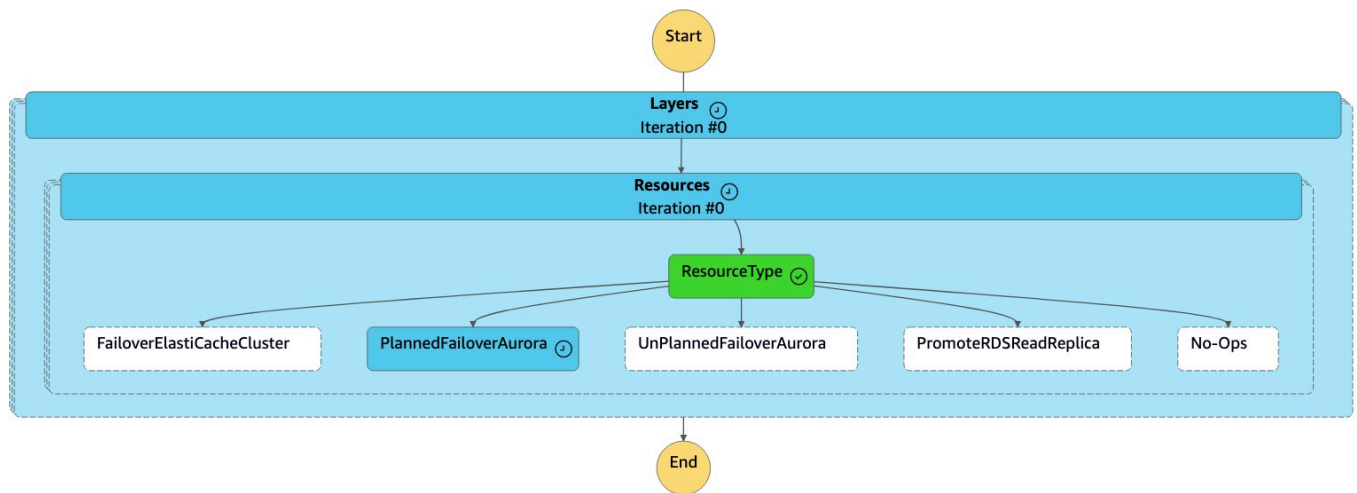
dr-orchestrator-stepfunction-FAILOVER状态机在事件状态期间被调用，以将您的主集群切换到您选择的辅助区域 (us-west-2)。

要执行切换，请执行以下操作：

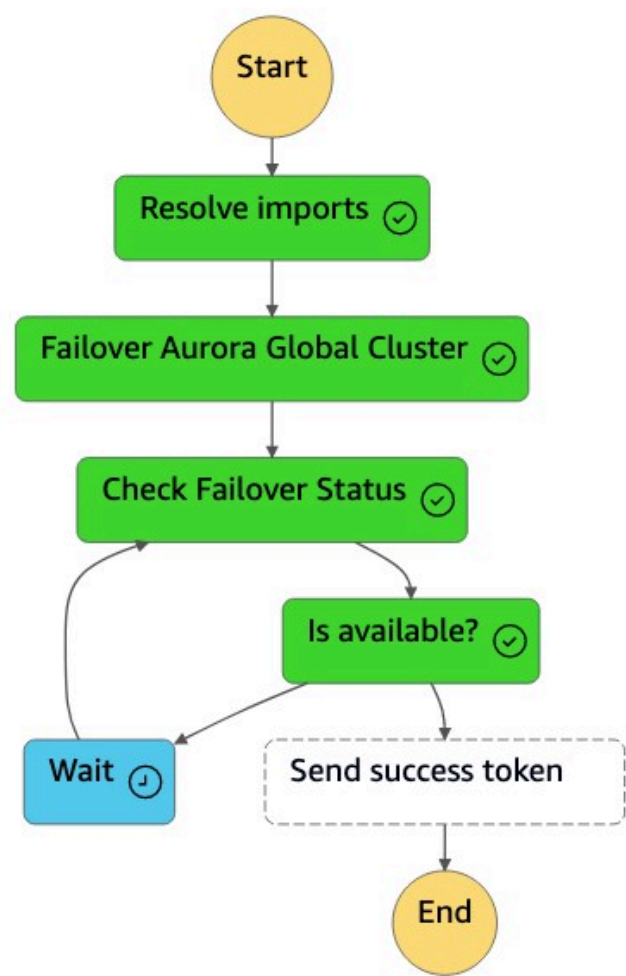
1. 登录到 AWS 管理控制台。
2. 将区域更改为 DR 区域 (us-west-2)。
3. 导航到服务，然后选择 Step Functions。
4. 导航到dr-orchestrator-stepfunction-FAILOVER状态机。
5. 选择“开始执行”，然后在该Input - optional部分中输入以下 JSON 代码：

```
{
  "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. dr-orchestrator-stepfunction-FAILOVER状态机以 PlannedFailoverAuroraMySQL 的形式读取资源类型，然后调用dr-orchestrator-stepfunction-planned-Aurora-failover状态机对 Aurora 全局数据库进行故障转移。



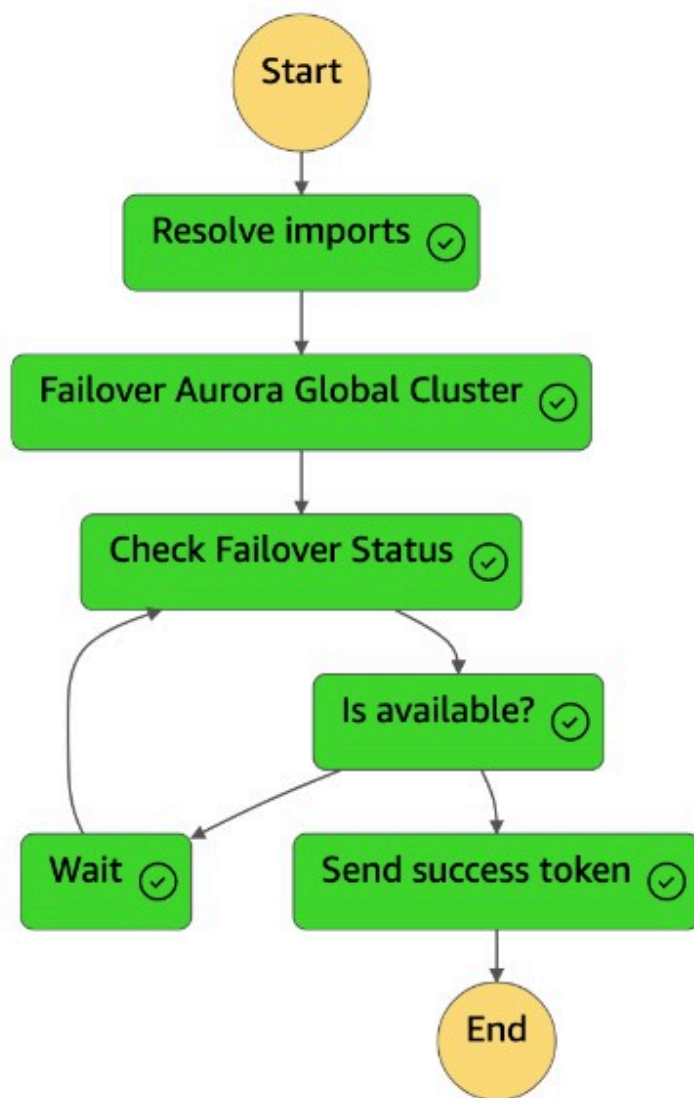
7. `dr-orchestrator-stepfunction-planned-Aurora-failover` 状态机执行以下步骤来切换 Aurora MySQL-Compatible 全局数据库角色。



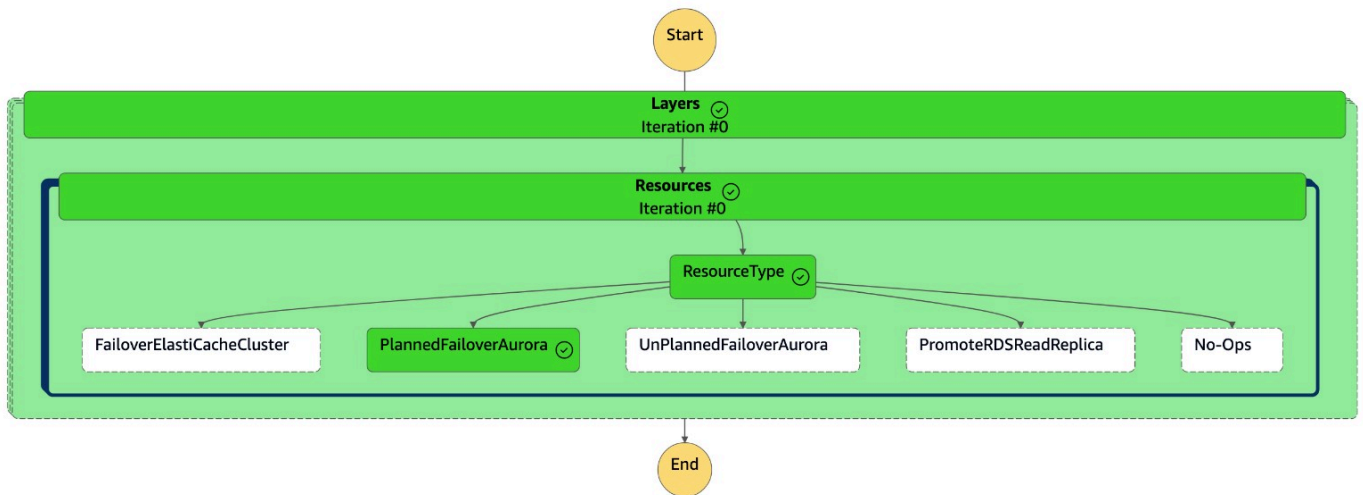
步骤	说明	期望值
解决导入问题	Lambda 函数用实际!Import <variable name>名称替换值。	"!Import dr-global db-cluster-mysql-global-identifier" 被替换为"dr-globaldb-cluster-mysql" 。

步骤	说明	期望值
故障转移 Aurora 全球集群	Lambda 函数调用 failover_global_cluster Boto3 API 来对 Aurora 全局数据库进行故障转移。	<pre>{ 'GlobalCluster': { 'GlobalClusterId entifier': 'dr-globa ldb-cluster-mysql', 'GlobalClusterRe sourceId': 'cluster- cce7f9bec2846db4', 'GlobalClusterArn': 'arn:aws:rds::xxx', 'Status': 'failing- over', } }</pre>
检查故障转移状态	Lambda 函数调用 describe_db_clusters Boto3 API 来检查故障转移的状态。	正在修改，可用
发送成功代币	Lambda 函数调用 send_task_success Boto3 API 并将成功令牌发送回状态机。DR Orchestrator Failover	h7XRiCdLtd/83p1E0dMccoxIzFhglsdkzpK9mBVKZsp7d9yrT1W

8. 导航至 Amazon RDS 控制台。在“状态”下，Aurora 全局数据库的值将从“可用”更改为“切换”或“正在修改”。
9. dr-orchestrator-stepfunction-planned-Aurora-failover状态机完成后，它会将成功令牌发送回dr-orchestrator-stepfunction-FAILOVER状态机。



10dr-orchestrator-stepfunction-FAILOVER状态机已完成。



在控制台上，辅助群集 (**dbcluster-02**) 的角色现在是主群集，并且该群集已准备好为 read/write 工作负载提供服务。原始主群集 (dbcluster-01) 的角色现在列为辅助群集。

手动计划外故障转移

在极少数情况下，Aurora 全球数据库的主数据库可能会出现意外中断 AWS 区域。如果发生这种情况，主 Aurora 数据库集群及其写入器节点将不可用，主集群和辅助集群之间的复制过程将停止。为了最大限度地减少停机时间 (RTO) 和数据丢失 (RPO)，请快速执行跨区域故障转移并重建 Aurora 全球数据库。有关更多信息，请参阅[从计划外中断中恢复 Amazon Aurora 全球数据库](#)。

执行计划外故障转移需要将辅助集群与 Aurora 全局数据库分离。在执行计划外故障转移之前，请停止在主 Aurora 数据库集群上写入应用程序。成功完成故障转移后，重新配置应用程序以写入新的主数据库群集。这种方法有助于防止数据丢失。如果主写入节点在故障转移过程中恢复联机，它还有助于避免数据不一致。

要执行计划外故障转移，请调用 `dr-orchestrator-stepfunction-FAILOVER` 状态机。在本示例中，辅助群集 (db-cluster-02) 在灾难恢复区域 (us-west-2) 中处于稳定状态。

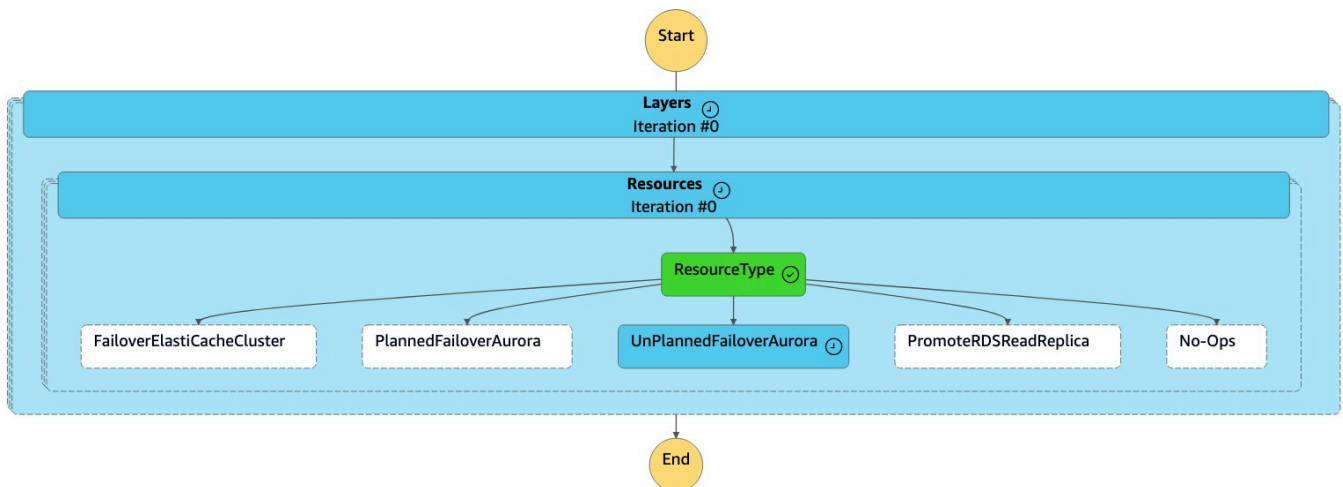
要执行故障转移，请执行以下操作：

1. 登录 控制台。
2. 将区域更改为 DR 区域 (us-west-2)。
3. 导航到服务，然后选择 Step Functions。
4. 导航到 `dr-orchestrator-stepfunction-FAILOVER` 状态机。

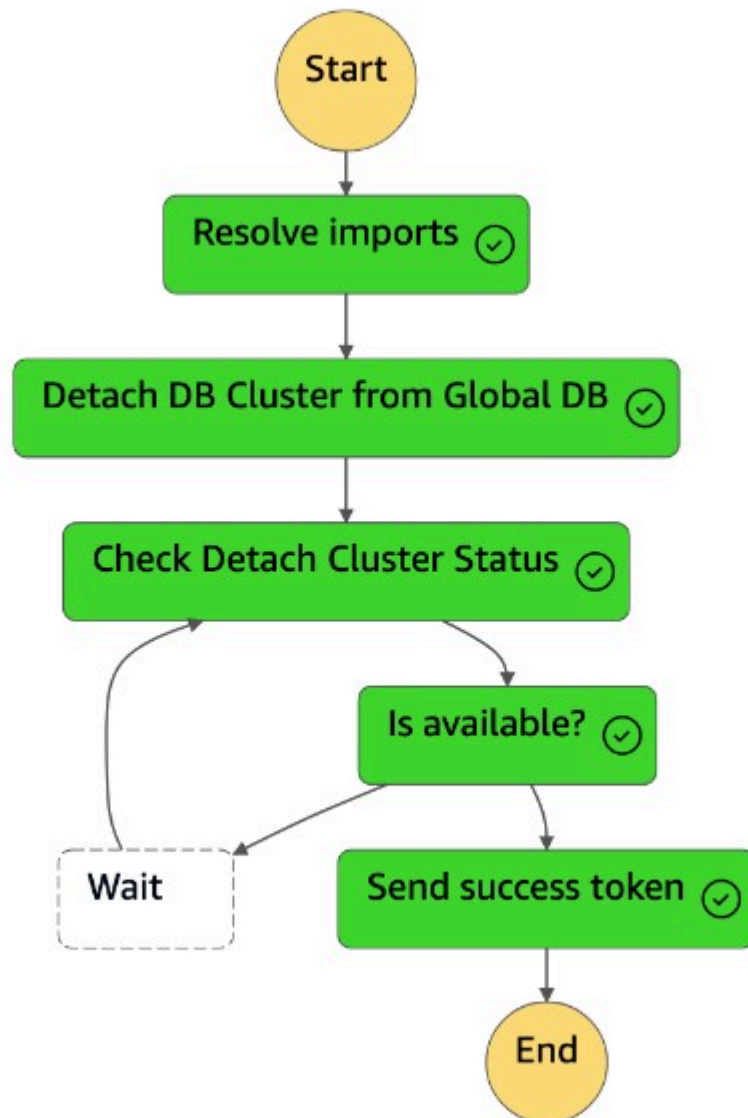
5. 选择“开始执行”，然后在该Input - optional部分中输入以下 JSON 代码，使用UnPlannedFailoverAurora作为 resourceType：

```
{
  "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. dr-orchestrator-stepfunction-FAILOVER状态机将资源类型读取为UnPlannedFailoverAuroraMySQL并Detach Cluster from Global Database从dr-orchestrator-stepfunction-unplanned-Aurora-failover状态机调用任务。

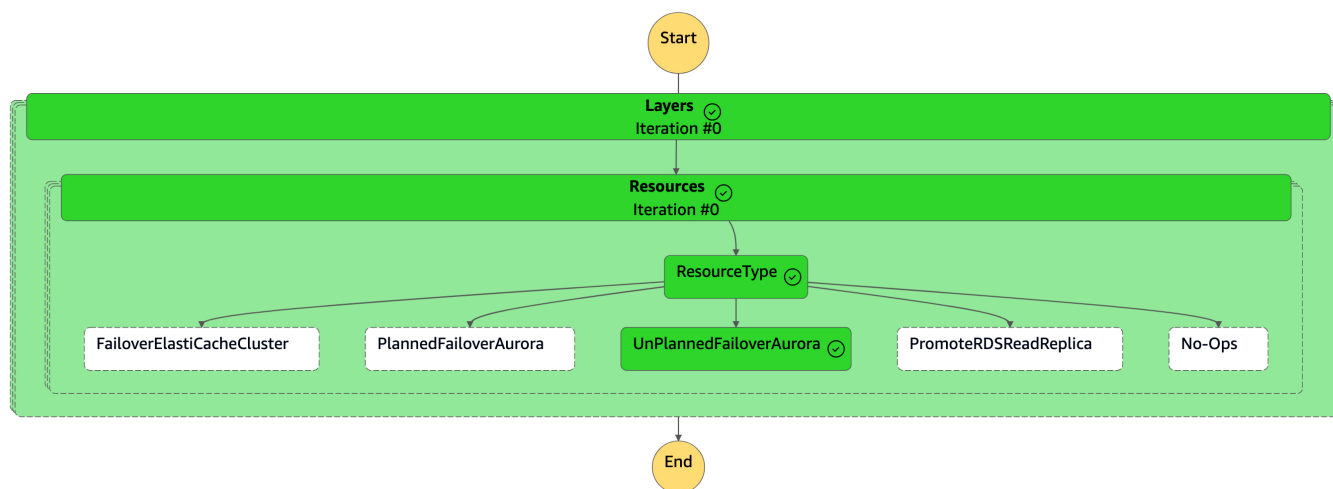


7. 该Detach Cluster from Global Database任务将辅助群集与全局数据库分离（删除）。



8. 辅助群集 (dbcluster-02) 已升级为独立群集，它可以为 read/write 工作负载提供服务。

9. dr-orchestrator-stepfunction-FAILOVER状态机已完成。



10 辅助集群 (dbcluster-02) 已从 Aurora 全局数据库分离，它将成为一个独立集群来 read/writer 处理工作负载。

11 重新配置应用程序，使用新的集群终端节点将所有写入操作发送到这个新的独立 Aurora 数据库集群。

故障恢复

灾难（或计划事件）解决后，故障恢复会将您的数据库返回到原始（或新的）主位置。计划外中断得到解决后，您可能需要将以前的主区域添加回 Aurora 全球数据库。您必须先从以前的主区域中删除现有数据库集群，从新的主区域创建新的数据库集群，然后使用托管计划内故障转移流程切换新集群的角色。

这可以被视为一项有计划的活动，您可以在非高峰时段或周末进行。

您必须手动 [修改 Amazon Aurora 数据库集群](#) 并禁用，然后才能从 DeletionProtection 以前的主区域 (us-east-1) 运行 DR Orchestrator FAILBACK 状态机，因为它是在创建 DeletionProtection。

DR Orchestrator Framework 使用 dr-orchestrator-stepfunction-FAILBACK 状态机自动执行删除现有集群并在以前的主区域中创建新集群的步骤。

要禁用 DeletionProtection，请执行以下操作：

1. 登录 控制台。
2. 将区域更改为以前的主区域 (us-east-1)。

3. 导航到 Amazon RDS 控制台，选择集群名称 (dbcluster-01)，然后选择修改。
4. 在“删除保护”下，清除“启用删除保护”复选框，然后选择“继续”。
5. 选择“立即应用”，然后选择“修改集群”。

DR Orchestrator FAILBACK状态机是在故障恢复过程中从以前的主区域 (us-east-1) 调用的。

要执行故障恢复，请执行以下操作：

1. 登录 控制台。
2. 将区域更改为以前的主区域 (us-east-1)。
3. 导航到“服务”，然后选择“Step Functions”。
4. 导航到DR Orchestrator FAILBACK状态机。
5. 选择“开始执行”，然后在该Input - optional部分中输入以下 JSON 代码：

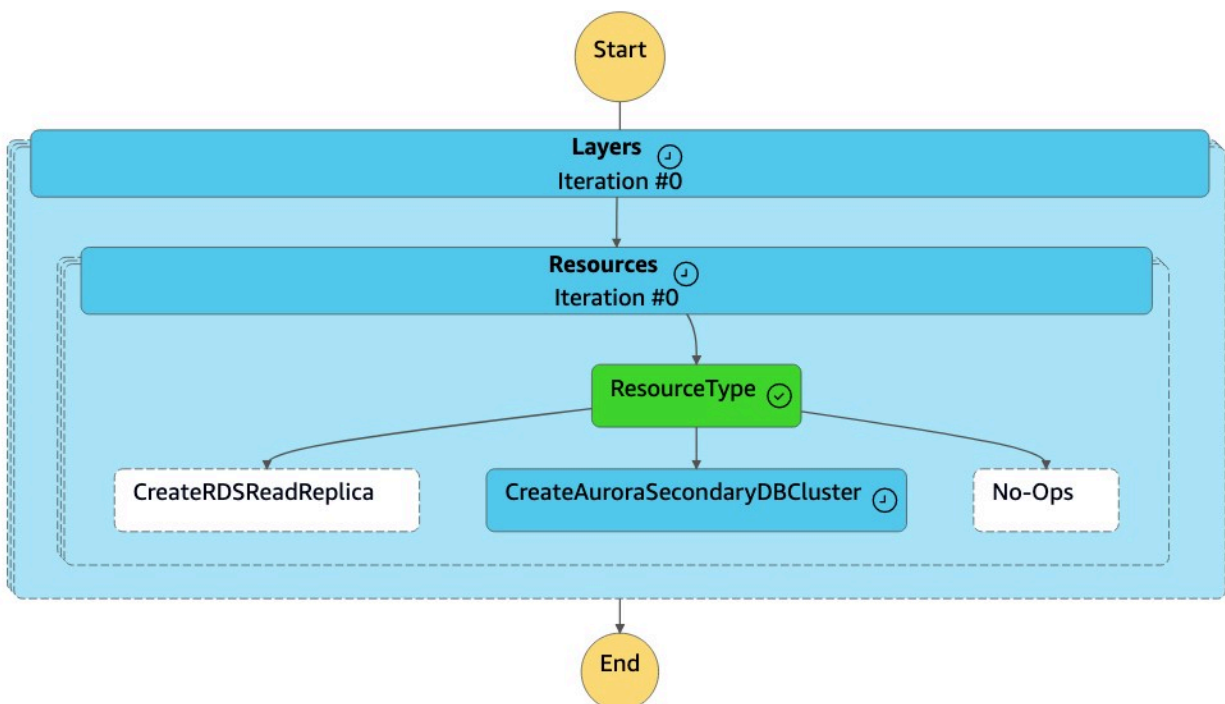
```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "CreateAuroraSecondaryDBCluster",
          "resourceName": "To create secondary Aurora MySQL Global Database Cluster",
          "parameters": {
            "GlobalClusterIdentifier": "!Import 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",
```

```

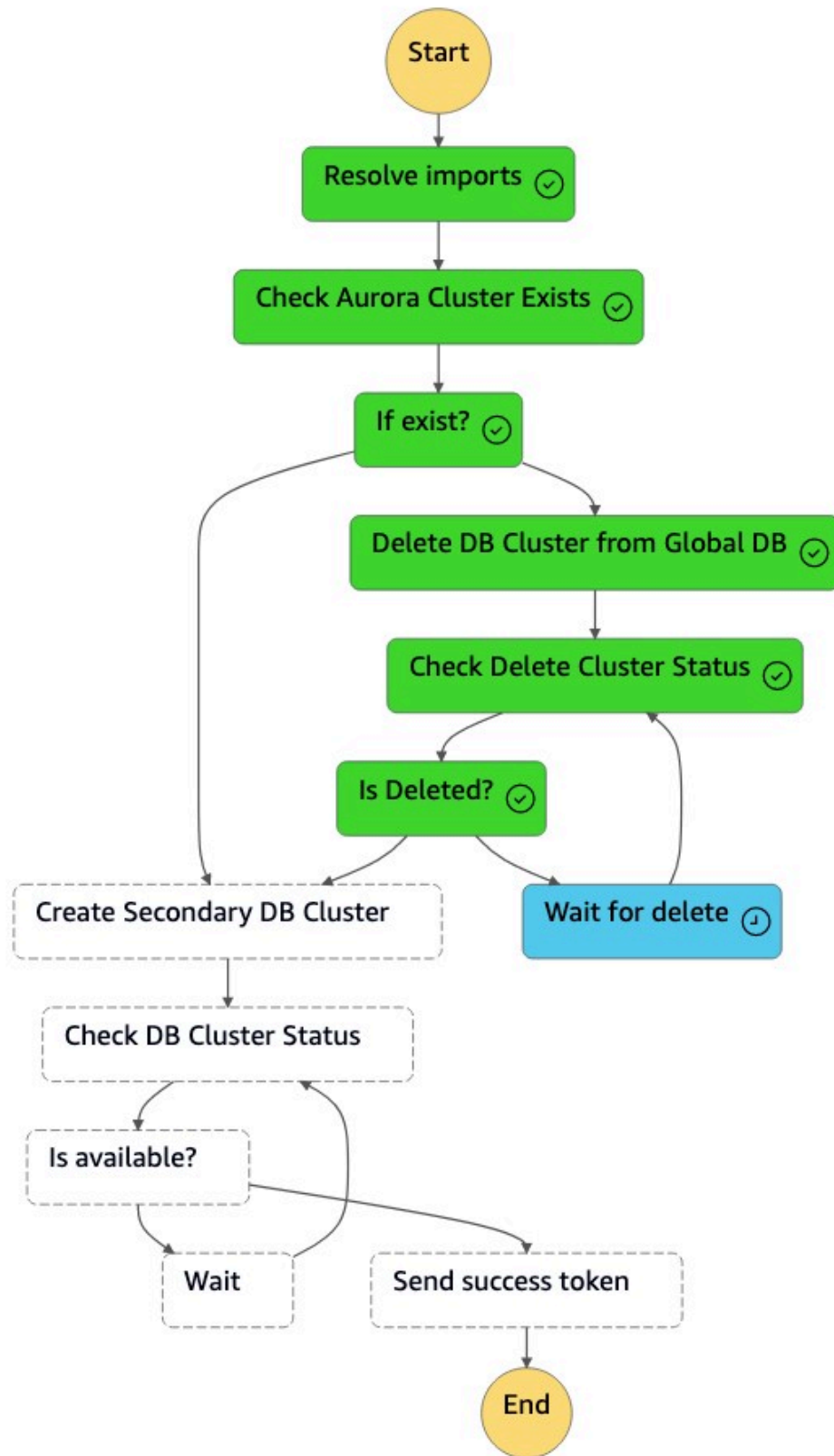
    "KmsKeyId": "!Import dr-globaldb-cluster-mysql-KmsKeyId",
    "SourceRegion": "!Import dr-globaldb-cluster-mysql-source-region",
    "ClusterRegion": "!Import dr-globaldb-cluster-mysql-cluster-region",
    "BackupRetentionPeriod": "7",
    "MonitoringInterval": "60",
    "StorageEncrypted": "True",
    "EnableIAMDatabaseAuthentication": "True",
    "DeletionProtection": "True",
    "CopyTagsToSnapshot": "True",
    "AutoMinorVersionUpgrade": "True",
    "MonitoringRoleArn": "!Import rds-mysql-instance-RDSMonitoringRole"
  }
}
]
}
}

```

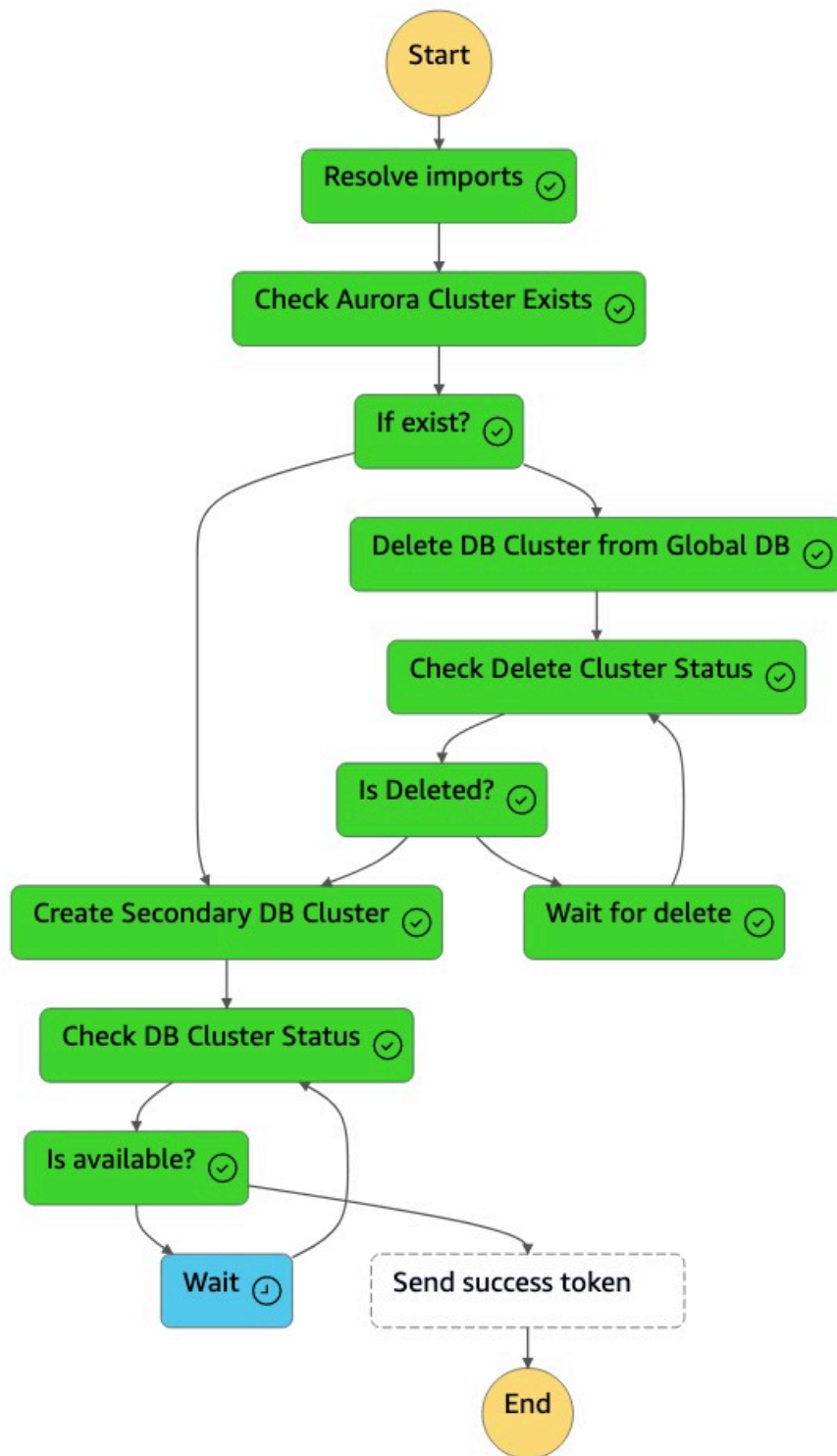
6. DR Orchestrator FAILBACK状态机将资源类型读为CreateAuroraSecondaryDBCluster，然后调用dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster状态机。



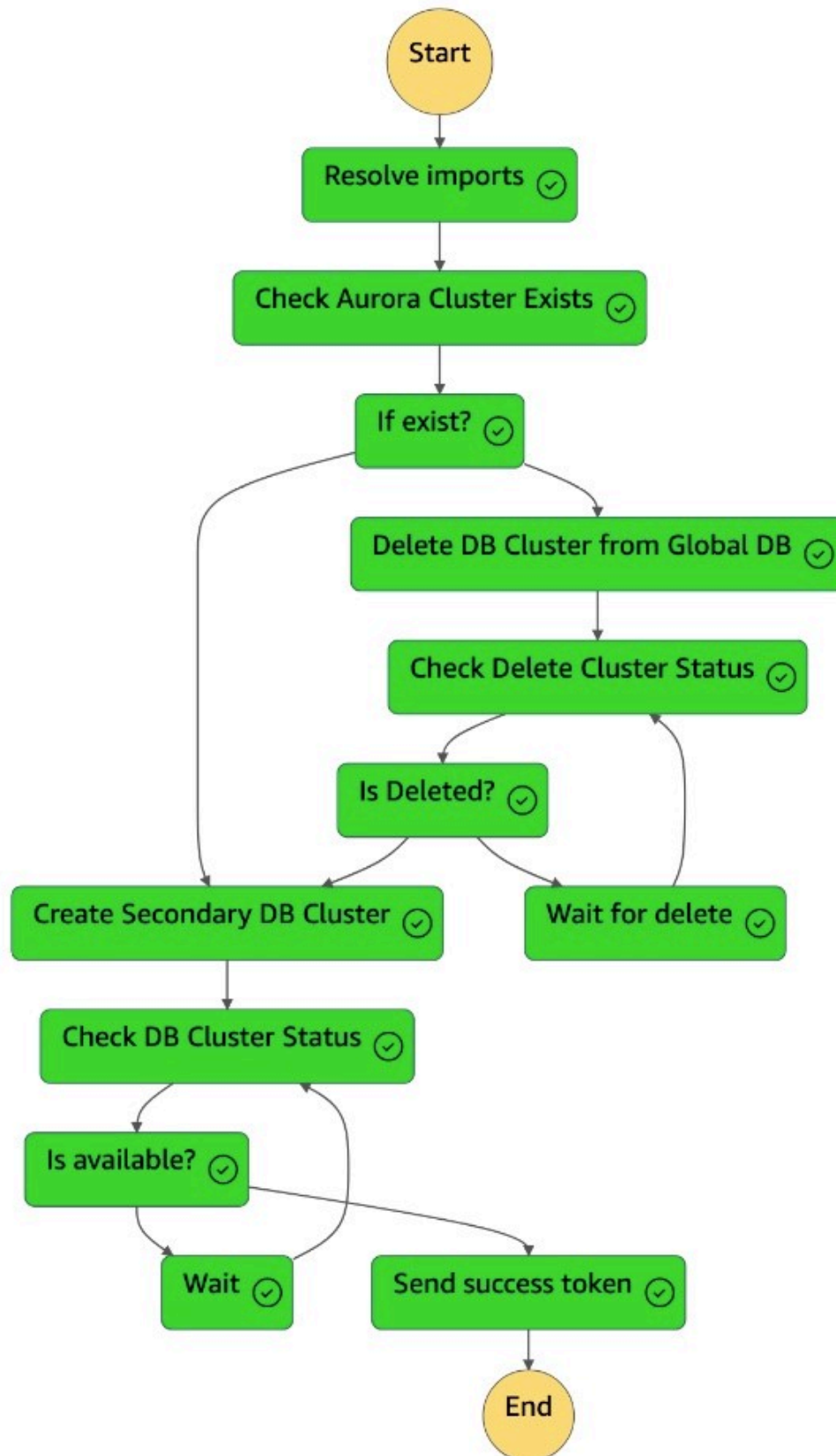
7. dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster状态机将现有集群 (dbcluster-01) 从以前的主区域 (us-east-1) 中删除。



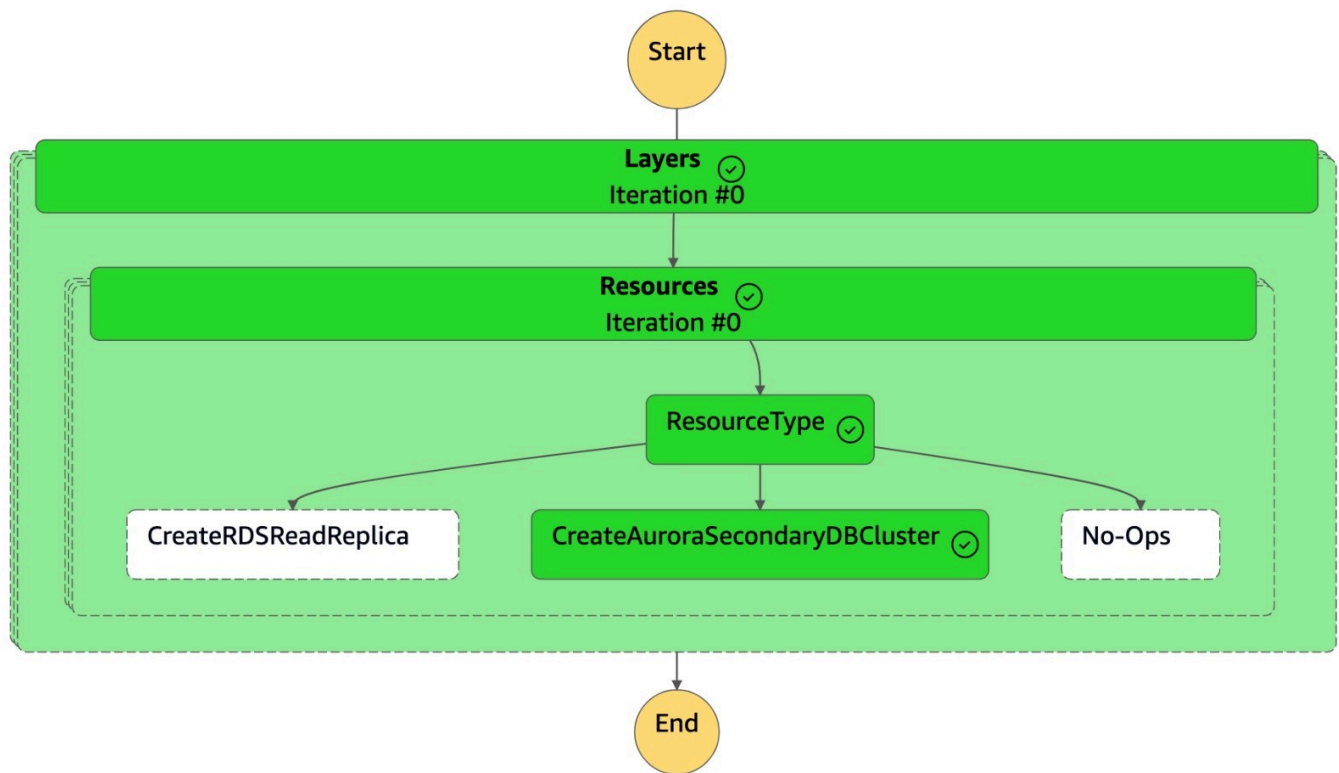
8. 删除集群 (dbcluster-01) 后，状态机将创建新的集群 (dbcluster-01) 和数据库实例，并作为辅助集群加入 Aurora 全局数据库，为只读工作负载提供服务。



9. 辅助群集可用后，`dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster`状态机就完成了，它会将成功令牌发送回DR Orchestrator Failback状态机。



10dr-orchestrator-stepfunction-FAILBACK状态机已完成。



11您可以在 Amazon RDS 控制台上验证 Aurora 全球数据库。

如果要将主数据库集群重新定位到 us-east-1，则可以按照切换部分中提到的步骤进行操作。

部署解决方案

设置先决条件后，请按照 [DR Orchestrator Framework](#) 模式中的说明进行操作。包含针对亚马逊和亚马逊RDS Aurora的说明 ElastiCache。要测试解决方案，请查看先决条件和限制，然后在上部署示例数据库 AWS。

先决条件

- 正在积极部署[AWS 账户](#)此解决方案。
- 创建[亚马逊简单存储服务 \(Amazon S3\) Service](#) 存储桶和模板中使用的资源的权限。[AWS CloudFormation](#)
- 在运行DR Orchestrator FAILBACK状态机之前：
 - 手动[修改RDS实例](#)或[修改 Amazon Aurora 数据库集群](#)，然后将其禁用DeletionProtection。
 - 如果您计划执行根本原因分析，请在删除 Amazon RDS 实例或 Aurora 集群之前[拍摄数据库快照](#) (RCA)。

限制

- 功能可用性和支持因每个数据库引擎的特定版本和不同版本而异 AWS 区域。有关跨区域复制功能和区域可用性的更多信息，请参阅[跨区域只读副本](#)。
- 此解决方案已使用带有一个只读副本的主数据库实例进行了测试。如果要使用多个只读副本，请在生产环境中实施解决方案之前对其进行全面测试。
- 对于亚马逊RDS，该解决方案仅支持以下引擎：[我的亚马逊RDSSQL](#)、[Postgre RDS SQL 的亚马逊](#)和 [MariaDB RDS 的亚马逊](#)。

在上部署示例数据库 AWS

要上演示数据库的跨区域灾难恢复解决方案 AWS，你可以部署 [Amazon RDS](#) for My SQL、[Amazon Aurora 我的SQL兼容版](#)和[亚马逊 ElastiCache \(RedisOSS\)](#)- 全球数据存储。使用 [aws-cross-region-dr-databases GitHub 存储库App-StackREADME](#)文件中的说明。在此示例中，主区域为us-east-1，次要区域为us-west-2。

应用程序堆栈使用“AWS CloudFormation [输出](#)”部分导出 DR Orchestrator Framework 用于自动执行数据库故障转移和故障恢复的参数。AWS

常见问题解答

查找有关用于故障转移和故障恢复的 DR Orchestrator 框架的问题的答案。

使用这种方法我能实现的 RPO 和 RTO 是什么？

有关恢复时间目标 (RTO) 和恢复点目标 (RPO) 的信息，请参阅中的[数据库灾难恢复策略策略](#)。AWS

必须使用 AWS CloudFormation 导出变量吗？

不行，您可以直接以 JSON 格式传递 Amazon Aurora 全球数据库或 Amazon RDS 数据库实例的值（例如：`- "RDSInstanceIdentifier": "rds-mysql-instance"`）。

我能否使用 DR Orchestrator 故障转移工作流程对多个 AWS 数据库进行故障切换？

是的，您可以在输入参数文件中传递多个资源来故障转移多个 AWS 数据库。以下代码示例显示了如何并行故障切换 Amazon RDS for MySQL 只读副本和亚马逊 ElastiCache（Redis OSS）全球数据存储库：

```
{
  "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"
    }
}
]
```

当我运行适用于 Amazon RDS 的 DR Orchestrator 故障恢复状态机时，如何避免出现 InvalidParameterCombination 错误？

错误的全文是：

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

为避免错误，请在运行DR Orchestrator FAILBACK状态机DeletionProtection之前通过禁用[RDS 实例来修改](#) RDS 实例。

后续步骤

在启动故障转移或故障恢复活动之前，您可以使用 DR Orchestrator Framework 添加批准流程。例如，在灾难恢复事件期间启动故障转移时，您可以在上面添加状态机，DR Orchestrator Failover 以便通过电子邮件发送[亚马逊简单通知服务 \(Amazon SNS\) Simple Notification Service 通知](#)。一旦获得批准，故障转移活动就会开始。

应用程序故障转移还可以与 DR Orchestrator Framework 集成，以切换 DNS 入口并将应用程序指向数据库实例或集群的新终端节点。

您可以使用[AWS X-Ray](#)跟踪来获取计算 RTO 的故障转移持续时间。您可以在 X-Ray 之上构建监控面板。

资源

- [跨 AWS 区域 使用全球数据存储进行复制](#) (亚马逊 ElastiCache (Redis OSS) 文档)
- [在 Amazon Aurora Global Database 中使用失效转移](#) (Aurora 文档)
- [自动管理 Amazon Aurora 全球数据库终端节点，实现计划内和计划外故障转移](#) (博客文章)
- [灾难恢复：使用 Terraform 为你的 Amazon Aurora 全球数据库提供 3 种故障转移场景 \(第 2 部分 \)](#) (博客文章)
- [数据库的灾难恢复策略 AWS](#) (AWS 规范性指导策略)
- [使用 DR Orchestrator 框架 \(AWS 规范性指导模式 \) 自动执行跨区域故障转移和故障恢复](#)

文档历史记录

下表介绍了本指南的一些重要更改。如果您希望收到有关未来更新的通知，可以订阅 [RSS 源](#)。

变更	说明	日期
初次发布	—	2024 年 5 月 3 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。