



在 上自動化關聯式資料庫的 DR 解決方案 AWS

# AWS 方案指引



# AWS 方案指引: 在上自動化關聯式資料庫的 DR 解決方案 AWS

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

|                                                                                              |        |
|----------------------------------------------------------------------------------------------|--------|
| 簡介 .....                                                                                     | 1      |
| 目標 .....                                                                                     | 1      |
| 概觀 .....                                                                                     | 2      |
| 容錯移轉程序 .....                                                                                 | 2      |
| DR Orchestrator FAILOVER 架構 .....                                                            | 2      |
| 容錯回復程序 .....                                                                                 | 4      |
| DR Orchestrator FAILBACK 架構 .....                                                            | 4      |
| DR 協調器失敗 .....                                                                               | 5      |
| 參數詳細資訊 .....                                                                                 | 6      |
| DR Orchestrator 失敗 .....                                                                     | 7      |
| Aurora 狀態和 Step Functions 狀態機器 .....                                                         | 10     |
| 穩定狀態 .....                                                                                   | 10     |
| 事件狀態 .....                                                                                   | 10     |
| 切換 .....                                                                                     | 10     |
| 手動意外容錯移轉 .....                                                                               | 16     |
| 容錯回復 .....                                                                                   | 19     |
| 部署解決方案 .....                                                                                 | 27     |
| 常見問答集 .....                                                                                  | 28     |
| 使用此方法可以達到什麼 RPO 和 RTO？ .....                                                                 | 28     |
| 是否必須使用 AWS CloudFormation 匯出變數？ .....                                                        | 28     |
| 我可以將 DR Orchestrator FAILOVER 工作流程來容錯移轉多個 AWS 資料庫嗎？ .....                                    | 28     |
| 執行 Amazon RDS 的 DR Orchestrator FAILBACK 狀態機器時，如何避免<br>InvalidParameterCombination 錯誤？ ..... | 29     |
| 後續步驟 .....                                                                                   | 30     |
| 資源 .....                                                                                     | 31     |
| 文件歷史紀錄 .....                                                                                 | 32     |
| .....                                                                                        | xxxiii |

# 自動化 上關聯式資料庫的 DR 解決方案 AWS

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

2024 年 5 月 ([文件歷史記錄](#))

在 Amazon Web Services ([上規劃災難復原 \(DR\) 策略](#)) 時，您可以為組織中的資料庫實作 DR 解決方案。AWS 您可以透過設定事件驅動架構來執行資料庫執行個體的容錯移轉，來自動化 DR 解決方案。此方法的優點包括下列項目：

- 減少手動協調所造成的人為錯誤
- 在某些情況下，提供最短的復原時間目標 (RTO)
- 促進 DR 解決方案的可重複測試

如果沒有自動化，主要 中的中斷 AWS 區域會需要組織中的資源手動執行執行手冊中的步驟。這些步驟可能包括多個程序，例如在次要 中提升 Amazon Relational Database Service (Amazon RDS) 僅供讀取複本 AWS 區域，以做為新的主要資料庫執行個體。AWS 提供[AWS 適用於 Python \(Boto3\) 的 SDK APIs](#) 來自動化這些動作，這可以形成 DR 解決方案的建置區塊，以滿足您的組織需求。

本指南討論集中式、資訊清單驅動的 DR 協調器，可協助您在相同 AWS 區域 內實現 AWS 資料庫的 DR 解決方案 AWS 帳戶。執行容錯移轉的決定仍需要人工介入，且超出自動化的範圍。

本指南的目標受眾包括應用程式架構師、基礎設施架構師、資料庫架構師和資料庫顧問。

## 目標

實作 DR Orchestrator Framework 解決方案可協助您達成下列結果：

- 在中斷期間減少手動錯誤 – 手動執行 DR 程序可能會容易發生錯誤。在 DR 活動期間自動化動作可減少手動錯誤，並有助於確保更順暢的容錯移轉或容錯回復。
- 大規模管理 DR – 如果您的組織擁有大量資料庫，建置 DR 自動化可協助您擴展 DR 解決方案。您可以使用 DR Orchestrator Framework 自動化 AWS 資料庫 DR 解決方案。DR Orchestrator 會使用 JSON 格式的參數檔案，自動執行 AWS 資料庫容錯移轉或容錯回復的步驟。

# DR Orchestrator Framework 概觀

DR Orchestrator Framework 提供一鍵式解決方案，可協調和自動化 AWS 資料庫的跨區域 DR。它使用 [AWS Step Functions](#) 和 [AWS Lambda](#) 在容錯移轉和容錯回復期間執行必要的步驟。Step Functions 狀態機器提供協調器設計中決策的基礎。用於執行容錯移轉或容錯回復動作的 API 操作會編碼為從狀態機器內呼叫的 Lambda 函數。Lambda 函數會執行 [適用於 Python \(Boto3\) 的 AWS SDK](#) APIs 來與 AWS 資料庫互動。

DR Orchestrator Framework 包含兩個主要狀態機器，對應至容錯移轉和容錯回復階段。

對於 Amazon RDS，容錯移轉階段會將跨區域 RDS 僅供讀取複本提升為獨立的資料庫執行個體。對於 Amazon Aurora，當主要區域在罕見且非預期的中斷期間關閉時，其寫入器節點無法使用。寫入器節點與次要叢集之間的複寫會停止。您必須從全域資料庫分離次要叢集，並將其提升為獨立叢集。應用程式可以連接並傳送寫入流量到獨立叢集。您可以使用這個相同的程序，將全域資料庫的主要資料庫叢集[切換](#)到次要區域。將此方法用於受控案例，如下所示：

- 操作維護
- 規劃的操作程序
- 將 Amazon ElastiCache (Redis OSS) 次要叢集提升為新的主要叢集

容錯回復階段會在即時主要區域和新的次要區域之間建立資料的即時複寫。

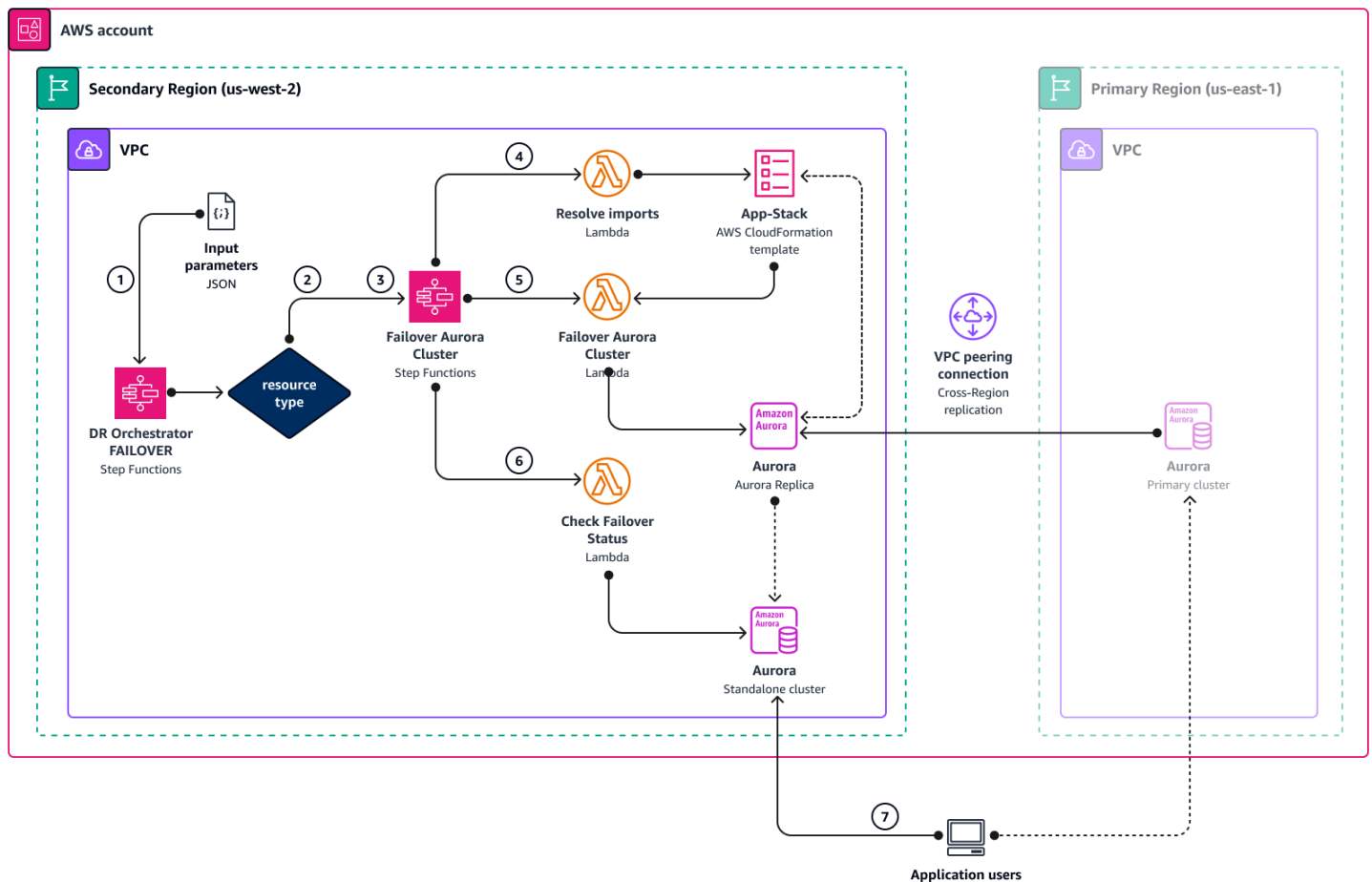
請務必了解 DR Orchestrator 僅適用於資料庫。參考這些資料庫且位於相同區域的所有應用程式，可能需要個別的串聯容錯移轉解決方案。在資料庫容錯移轉至次要區域後，需要更新應用程式以連線至新的資料庫執行個體，該執行個體將做為資料來源。

## 容錯移轉程序

若要執行容錯移轉，請執行 DR Orchestrator FAILOVER 狀態機器。在此階段，次要資料庫已存在於次要區域中，可以是僅供讀取複本 (Amazon RDS) 或次要叢集 (Amazon Aurora)。當您執行 DR Orchestrator FAILOVER 狀態機器時，它會提升次要資料庫成為主要資料庫。

## DR Orchestrator FAILOVER 架構

下圖顯示使用 DR Orchestrator 時 Amazon Aurora 容錯移轉程序的概念。Amazon Aurora 和 Amazon ElastiCache 使用相同的工作流程，但使用不同的狀態機器和 Lambda 函數。



1. DR Orchestrator FAILOVER 狀態機器會讀取輸入 JSON 參數。
2. 根據 resourceType 參數，狀態機器會呼叫其他 狀態機器：Failover Aurora Cluster、Promote RDS Read Replica或 Failover ElastiCache。如果輸入中傳遞多個資源，這些狀態機器會平行執行。
3. Failover Aurora Cluster 狀態機器會在下列三個步驟中呼叫 Lambda 函數。
4. Resolve imports Lambda 函數"! import <export-variable-name>"會以App-Stack AWS CloudFormation 範本中的實際值解析。
5. Lambda Failover Aurora Cluster 函數會將僅供讀取複本 提升為獨立資料庫執行個體。
6. Lambda Check Failover Status函數會檢查提升資料庫執行個體的状态。狀態為可用後，Lambda 函數會將成功字符傳回給呼叫狀態機器並完成。
7. 您可以將應用程式重新導向至 DR 區域 (us-west-2) 中的獨立資料庫，該區域現在是主要資料庫。

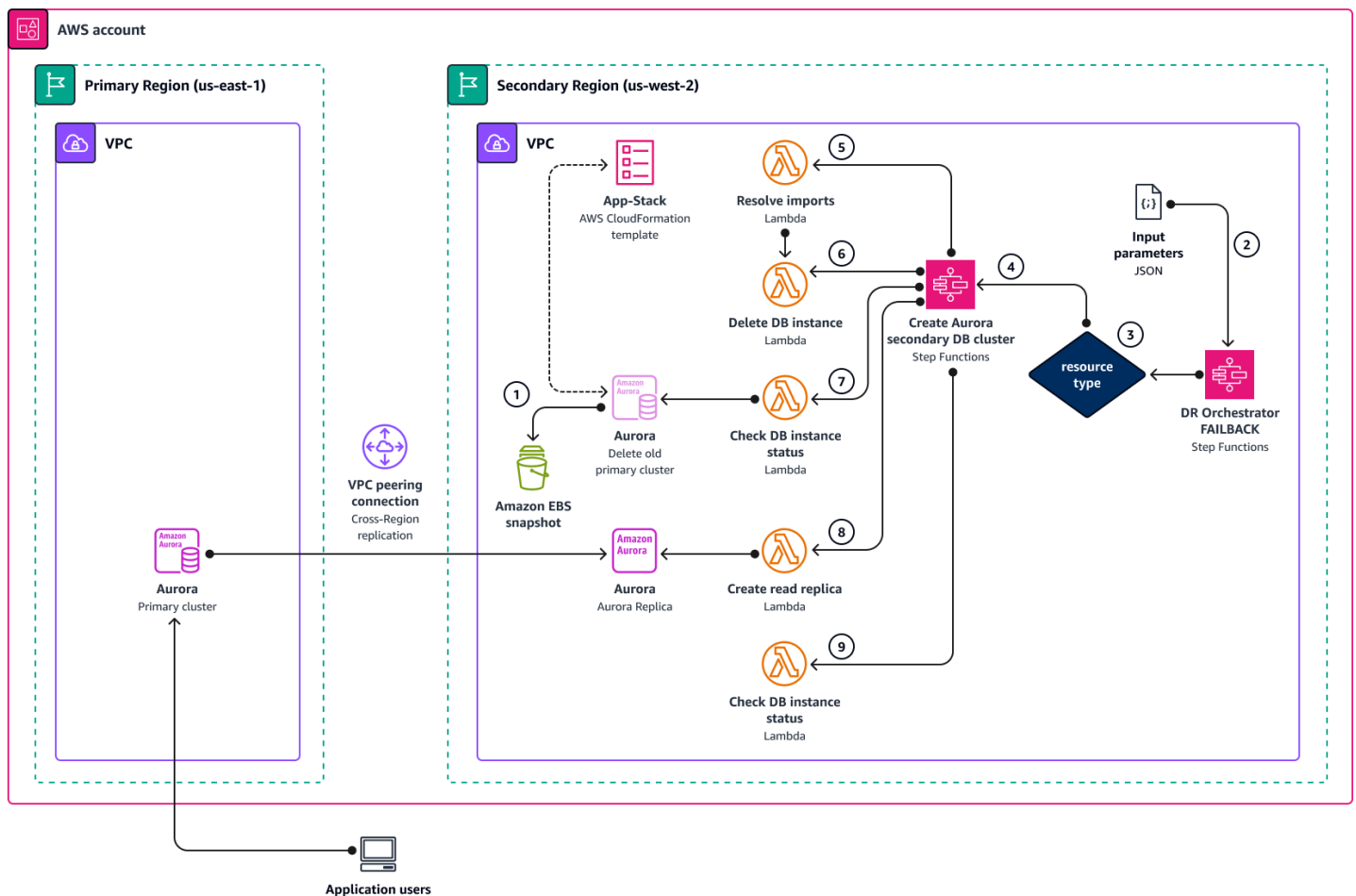
## 容錯回復程序

在您的前一個主要區域 (us-east-1) 再次啟動後，您可以容錯移轉回該區域，讓其中的資料庫再次 us-east-1 成為主要區域。若要啟動容錯回復，請執行 DR Orchestrator FAILBACK 狀態機器。如名稱所示，此狀態機器會開始將新主要區域 (us-west-2) 中的變更複寫回先前的主要區域 (us-east-1)，該區域做為目前的次要區域。

在兩個區域之間建立複寫之後，您可以啟動容錯回復。若要容錯回復並返回原始主要區域 (us-east-1)，請在目前的次要區域 (us-east-1) 中執行 DR Orchestrator FAILOVER 狀態機器，將其提升為主要區域。

## DR Orchestrator FAILBACK 架構

下圖顯示使用 DR Orchestrator 時 Amazon Aurora 的容錯回復程序概念。



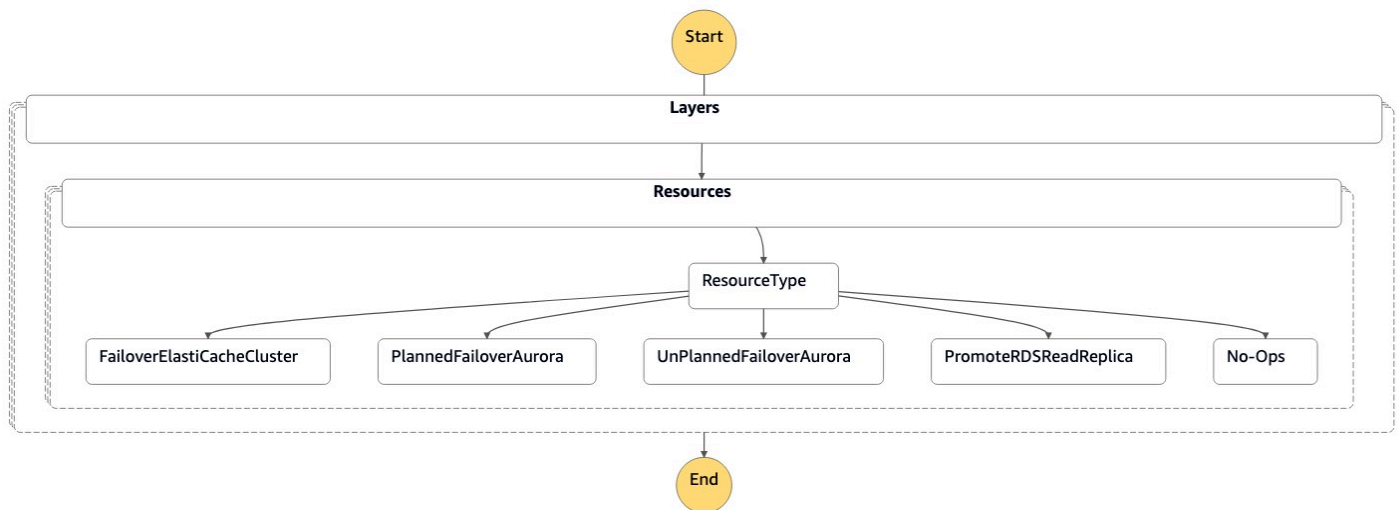
1. 開始容錯回復之前，請手動建立資料庫快照，以便在執行根本原因分析 (RCA) 時使用。

- 此外，請停用上一個主要區域 ( ) 中 Aurora 叢集 DeletionProtection 的 us-east-1。
2. DR Orchestrator FAILBACK 狀態機器會讀取輸入 JSON 參數。
  3. 根據 resourceType, DR Orchestrator FAILBACK 狀態機器會呼叫 Create Aurora Secondary DB Cluster 狀態機器。
  4. Create Aurora Secondary DB Cluster 狀態機器會在下列五個步驟中呼叫 Lambda 函數。
  5. Resolve import Lambda 函數"! import <export-variable-name>"會以 App-Stack CloudFormation 範本中的實際值解析。
  6. Delete DB Instance Lambda 函數會刪除先前的主要執行個體。
  7. Check DB instance status Lambda 函數會檢查, Delete DB Instance status 直到刪除資料庫為止。
  8. Create Read Replica Lambda 函數會從位於新主要區域的資料庫執行個體, 在次要區域中建立僅供讀取複本。
  9. Check DB instance status Lambda 函數會檢查僅供讀取複本資料庫執行個體狀態。當狀態為可用時, Lambda 函數會將成功字符傳回至呼叫狀態機器, 該機器已完成。

## DR 協調器失敗

在主要區域 (us-east-1) 關閉時或在營運維護等計劃事件期間, 在 DR 事件中使用 DR Orchestrator FAILOVER 狀態機器。

您可以呼叫 函數, 透過單一或多個資料庫平行失敗。



狀態機器接受 JSON 格式的參數, 如下列程式碼所示:

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

## 參數詳細資訊

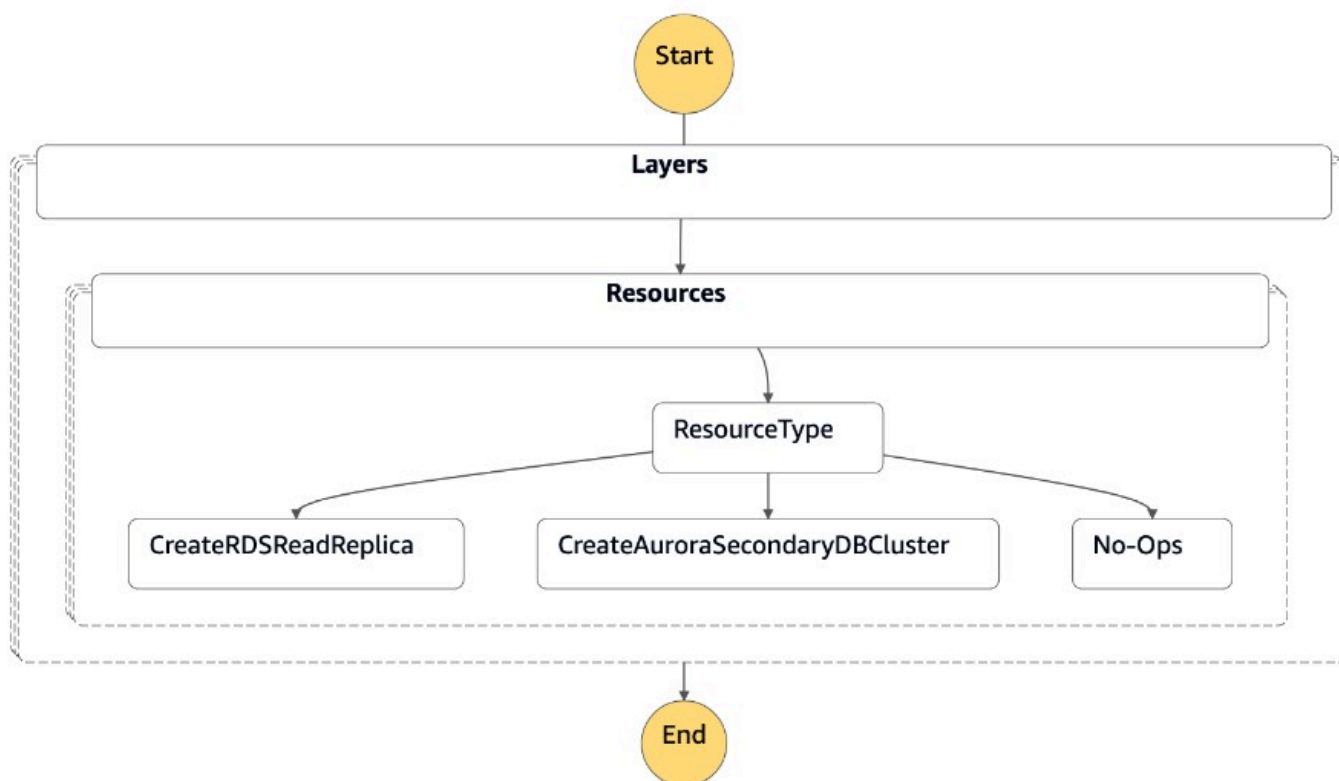
下表顯示DR Orchestrator FAILOVER狀態機器使用的參數。

| 參數名稱            | 說明                                    | 預期值          |
|-----------------|---------------------------------------|--------------|
| layer ( 必要：數字 ) | 處理序列。在執行第 2 層資源之前，必須先執行第 1 層中定義的所有資源。 | 1 或 2，以此類推   |
| 資源 ( 必要：字典陣列 )  | 單一 layer 內的所有資源都會平行執行。                | <div>{</div> |

| 參數名稱                 | 說明                       | 預期值                                                                                                                                     |
|----------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|                      |                          | <pre>"resourceType": "String", "resourceName": "String", "parameters": { "&lt;param1&gt;": "&lt;!Import cft-output-1"&gt;, .... }</pre> |
| resourceType (必要：字串) | 用於識別資源的資源類型              | PromoteRDSReadReplica 或 FailoverElastiCacheCluster                                                                                      |
| resourceName (選用：字串) | 識別這些資源所屬的應用程式產品組合        | Promote RDS for MySQL Read Replica                                                                                                      |
| 參數 (必要：字典陣列)         | 容錯移轉或容錯回復 AWS 資料庫所需的參數清單 | <pre>{     "&lt;param1&gt;": "&lt;!Import cft-output-1"&gt;,     "&lt;param2&gt;": "&lt;!Import cft-output-2"&gt;, }</pre>              |

## DR Orchestrator 失敗

當前一個主要區域 (us-east-1) 啟動時，在 DR 事件之後使用 DR Orchestrator FAILBACK 狀態機器。您可以從新的主要區域 (us-west-2) 在先前主要區域中建立 Amazon RDS 的僅供[讀取複本](#)，以符合您的 DR 策略。由於這是計劃的事件，因此您可以在週末或離峰上班時間排程此活動，並估計停機時間。



狀態機器接受 JSON 格式的參數，如下列程式碼所示：

```

{
  "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 狀態和 Step Functions 狀態機器

本節涵蓋容錯移轉和回復 Amazon Aurora 叢集的特定程序和狀態機器。叢集會設定為全域資料庫。

## Note

基於示範目的，此範例使用 Aurora MySQL 相容版本。您可以針對 Aurora PostgreSQL 相容版本使用類似的步驟。

## 穩定狀態

在穩定狀態下，已使用兩個資料庫叢集建立 Amazon Aurora MySQL 相容全域資料庫 (dr-globaldb-cluster-mysql)。第一個資料庫叢集 (db-cluster-01) 已在主要 AWS 區域 (us-east-1) 中建立，以提供讀取/寫入工作負載。第二個資料庫叢集 (db-cluster-02) 已在次要區域 (us-west-2) 中建立，以伺服器唯讀工作負載。

除了提供 DR 解決方案之外，您還可以將讀取查詢從應用程式路由到次要資料庫叢集，以減少主要資料庫叢集的負載。每個叢集分別包含一個資料庫執行個體 dbcluster-02-usw2-instance-2，稱為 dbcluster-01-use1-instance-1 和。

## 事件狀態

透過使用 Amazon Aurora 全域資料庫，您可以快速規劃災難並從中復原。災難復原通常使用復原時間目標 (RTO) 和復原點目標 (RPO) 的值進行測量。如需詳細資訊，請參閱 [在 Amazon Aurora 全域資料庫中使用切換或容錯移轉](#)。

使用 Aurora 全域資料庫時，有兩種不同的容錯移轉方法：

- 切換（受管計劃容錯移轉）
- 容錯移轉（手動意外容錯移轉，或分離和提升）

## 切換

切換適用於受控環境，例如操作維護和其他規劃的操作程序。透過使用受管計劃容錯移轉，您可以將 Aurora 全域資料庫的主要資料庫叢集重新定位至其中一個次要區域。由於切換會等待次要資料庫叢集

與主要資料庫同步，因此 RPO 為 0（無資料遺失）。若要進一步了解，請參閱[執行 Amazon Aurora 全域資料庫的切換](#)。

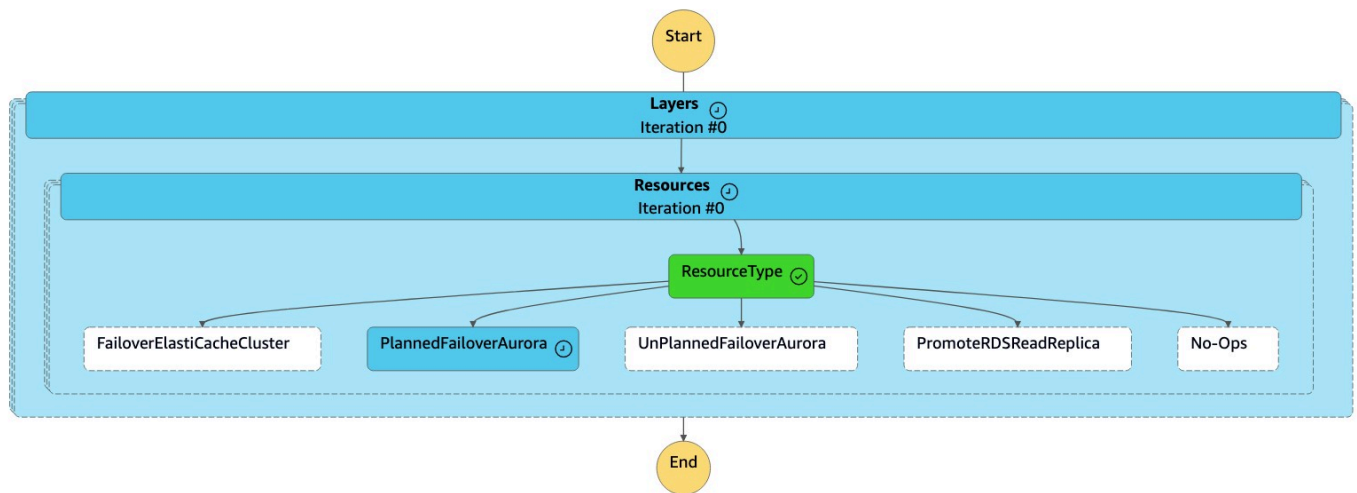
狀態dr-orchestrator-stepfunction-FAILOVER機器會在事件狀態期間叫用，將您的主要叢集切換到您選擇的次要區域(us-west-2)。

若要執行切換，請執行下列動作：

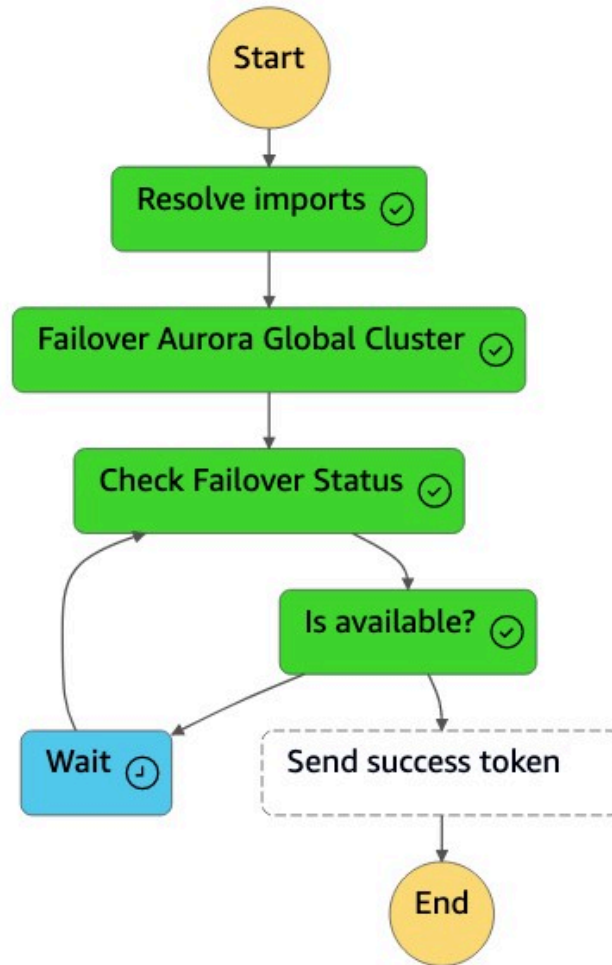
1. 登入 AWS 管理主控台。
2. 將區域變更為 DR 區域(us-west-2)。
3. 導覽至 服務，然後選擇 Step Functions。
4. 導覽至dr-orchestrator-stepfunction-FAILOVER狀態機器。
5. 選擇開始執行，然後在 Input - optional區段中輸入下列 JSON 程式碼：

```
{
  "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 狀態機器會將資源類型讀取為PlannedFailoverAuroraMySQL，並呼叫dr-orchestrator-stepfunction-planned-Aurora-failover狀態機器以容錯移轉 Aurora 全域資料庫。



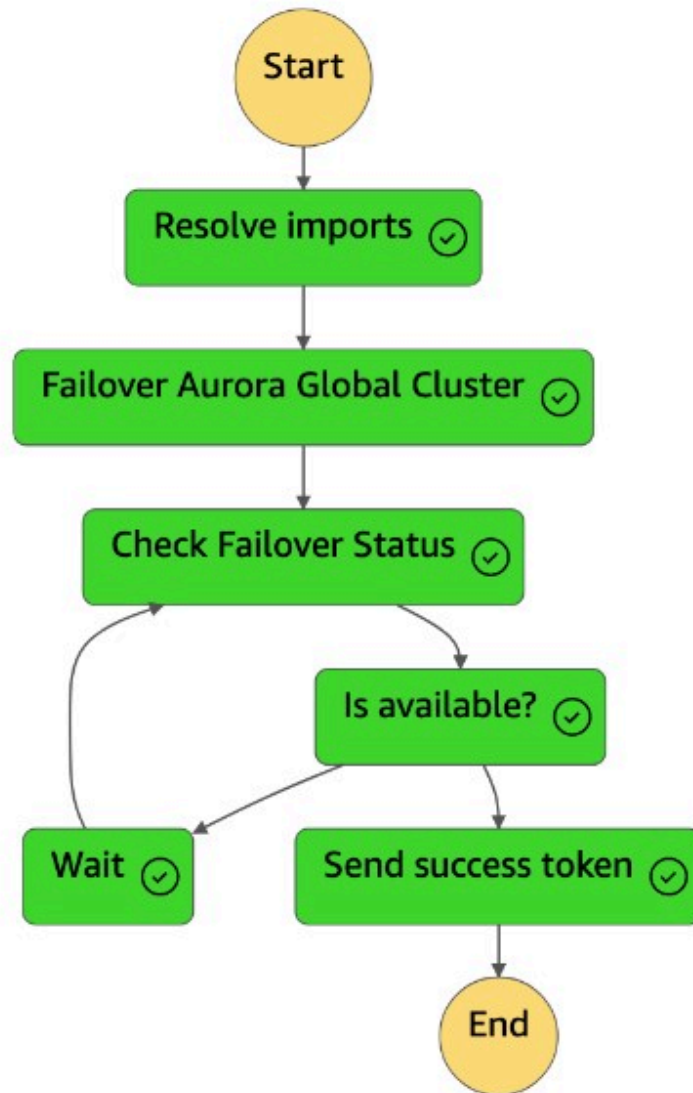
7. `dr-orchestrator-stepfunction-planned-Aurora-failover` 狀態機器會執行下列步驟，以切換 Aurora MySQL 相容全域資料庫角色。



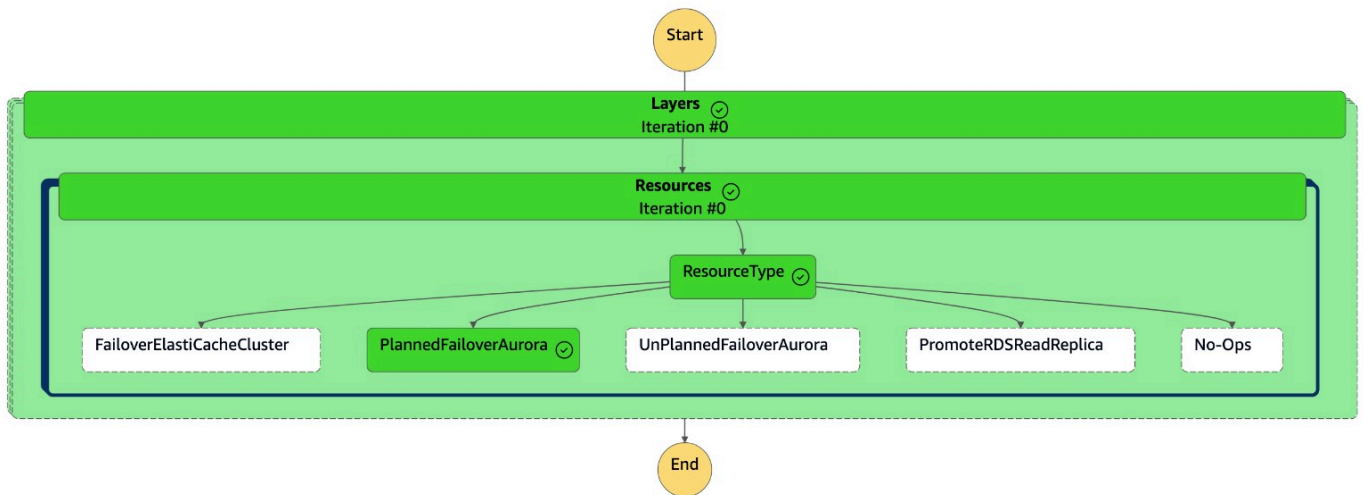
| 步驟   | 說明                                          | 預期值                                                                                      |
|------|---------------------------------------------|------------------------------------------------------------------------------------------|
| 解決匯入 | Lambda 函數會將!Import <variable name>值取代為實際名稱。 | "!Import dr-global db-cluster-mysql-global-identifier" 已由取代"dr-globaldb-cluster-mysql" 。 |

| 步驟               | 說明                                                                                              | 預期值                                                                                                                                                                                                                                                                   |
|------------------|-------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 容錯移轉 Aurora 全域叢集 | Lambda 函數會呼叫 <a href="#">failover_global_cluster</a> Boto3 APIs，以容錯移轉 Aurora 全域資料庫。             | <pre>{   'GlobalCluster': {     'GlobalClusterId entifier': 'dr-globa ldb-cluster-mysql',     'GlobalClusterRe sourceId': 'cluster- cce7f9bec2846db4',     'GlobalClusterArn':       'arn:aws:rds::xxx',     'Status': 'failing- over',     ....     ....   } }</pre> |
| 檢查容錯移轉狀態         | Lambda 函數會呼叫 <a href="#">describe_db_clusters</a> Boto3 APIs來檢查容錯移轉的狀態。                         | 修改，可用                                                                                                                                                                                                                                                                 |
| 傳送成功權杖           | Lambda 函數會呼叫 <a href="#">send_task_success</a> Boto3 APIs，並將成功字符傳回DR Orchestrator Failover狀態機器。 | h7XRiCdLtd/83p1E0d<br>MccoxIzFhglsdkzpK9<br>mBVKZsp7d9yrT1W                                                                                                                                                                                                           |

- 導覽至 Amazon RDS 主控台。在狀態下，Aurora 全域資料庫的值會從可用變更為切換或修改。
- dr-orchestrator-stepfunction-planned-Aurora-failover 狀態機器完成後，它會將成功字符傳回dr-orchestrator-stepfunction-FAILOVER狀態機器。



10dr-orchestrator-stepfunction-FAILOVER 狀態機器已完成。



在主控台上，次要叢集 (dbcluster-02) 的角色現在是主要叢集，而且叢集已準備好提供讀取/寫入工作負載。原始主要叢集 (dbcluster-01) 的角色現在列為次要叢集。

## 手動意外容錯移轉

在極少數情況下，您的 Aurora 全域資料庫可能會在其主要資料庫發生意外中斷 AWS 區域。如果發生此情況，您的主要 Aurora 資料庫叢集及其寫入器節點將無法使用，而主要叢集和次要叢集之間的複寫則會停止。為了將停機時間 (RTO) 和資料遺失 (RPO) 降至最低，會快速執行跨區域容錯移轉，並重建您的 Aurora 全域資料庫。如需詳細資訊，請參閱[從意外中斷復原 Amazon Aurora 全域資料庫](#)。

執行意外容錯移轉需要您將次要叢集與 Aurora 全域資料庫分離。在執行意外容錯移轉之前，請先停止主要 Aurora 資料庫叢集上的應用程式寫入。成功完成容錯移轉後，請重新設定應用程式以寫入新的主要資料庫叢集。此方法有助於防止資料遺失。如果主要寫入器節點在容錯移轉程序期間恢復上線，也有助於避免資料不一致。

若要執行意外容錯移轉，請呼叫 dr-orchestrator-stepfunction-FAILOVER 狀態機器。在此範例中，次要叢集 (db-cluster-02) 位於穩定狀態的 DR 區域 (us-west-2)。

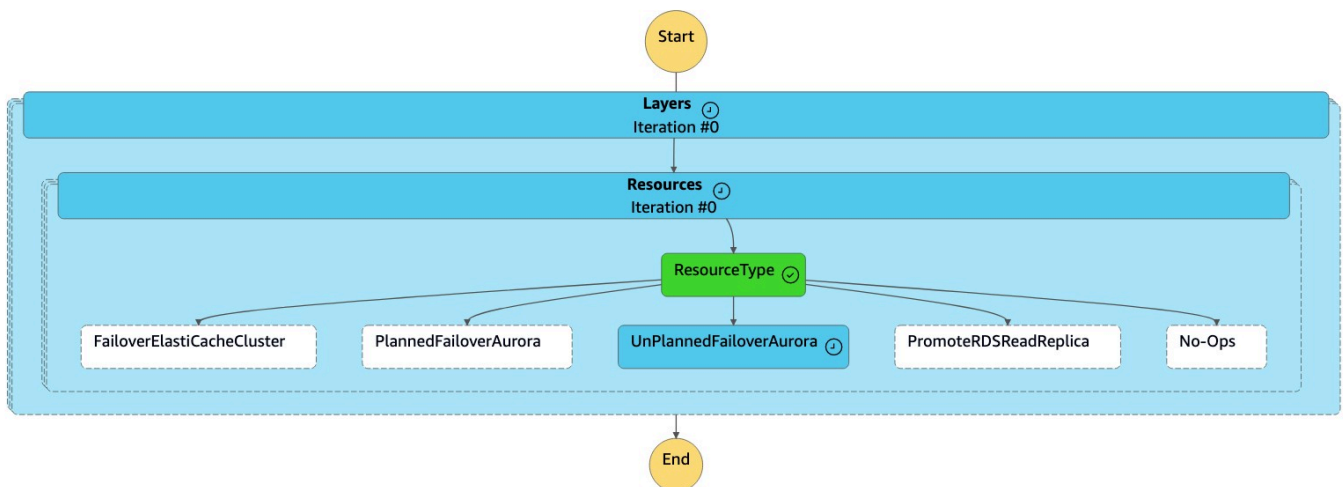
若要執行容錯移轉，請執行下列動作：

1. 登入 主控台。
2. 將 區域變更為 DR 區域 (us-west-2)。
3. 導覽至 服務，然後選擇 Step Functions。
4. 導覽至 dr-orchestrator-stepfunction-FAILOVER 狀態機器。

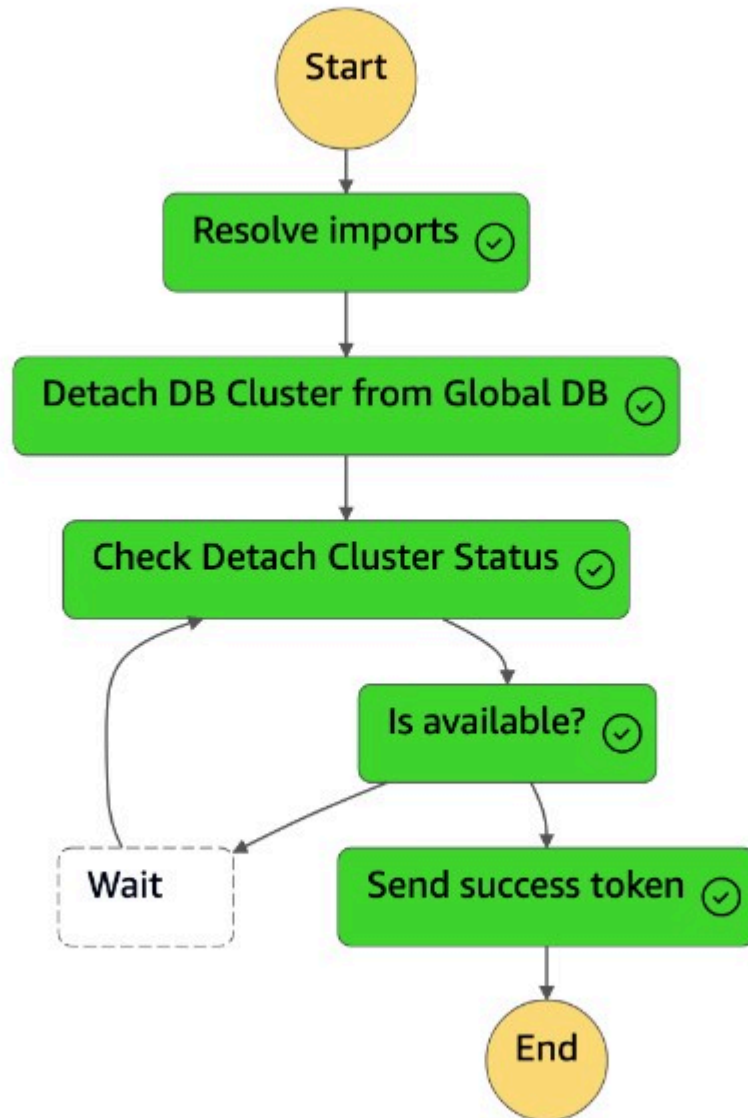
5. 選擇開始執行，然後在 Input - optional 區段中輸入下列 JSON 程式碼，使用 UnPlannedFailoverAurora 做為 resourceType：

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

6. dr-orchestrator-stepfunction-FAILOVER 狀態機器會將資源類型讀取為 UnPlannedFailoverAuroraMySQL 並從 Detach Cluster from Global Database dr-orchestrator-stepfunction-unplanned-Aurora-failover 狀態機器呼叫任務。

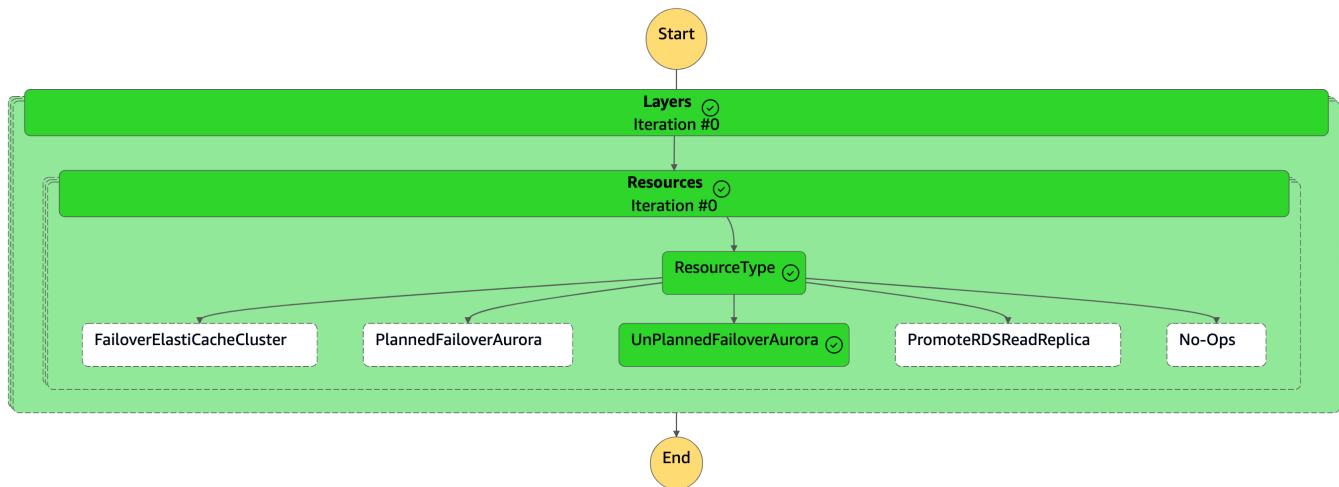


7. Detach Cluster from Global Database 任務會從全域資料庫分離（移除）次要叢集。



8. 次要叢集 (dbcluster-02) 已提升為獨立叢集，並可處理讀取/寫入工作負載。

9. dr-orchestrator-stepfunction-FAILOVER 狀態機器已完成。



10次要叢集 (dbcluster-02) 與 Aurora 全域資料庫分離，並且會成為獨立的叢集，為讀取/寫入器工作負載提供服務。

11重新設定您的應用程式，使用其新的叢集端點將所有寫入操作傳送至此新的獨立 Aurora 資料庫叢集。

## 容錯回復

故障恢復會在災難（或排程事件）解決後，將您的資料庫傳回原始（或新的）主要位置。當意外中斷解決後，您可能想要將之前的主要區域新增回 Aurora 全球資料庫。您必須先從先前的主要區域刪除現有的資料庫叢集，從新的主要區域建立新的資料庫叢集，然後使用受管計劃容錯移轉程序來切換新叢集的角色。

這可以視為您可以在離峰時間或週末執行的計劃活動。

您必須手動[修改 Amazon Aurora 資料庫叢集](#)，並在從先前的主要區域 (us-east-1) 執行DR Orchestrator FAILBACK狀態機器DeletionProtection之前停用，因為它是使用建立的DeletionProtection。

DR Orchestrator Framework 使用 dr-orchestrator-stepfunction-FAILBACK 狀態機器自動執行刪除現有叢集的步驟，並在先前的主要區域中建立新的叢集。

若要停用 DeletionProtection，請執行下列動作：

1. 登入 主控台。
2. 將 區域變更為先前的主要區域 (us-east-1)。
3. 導覽至 Amazon RDS 主控台，選取叢集名稱 (dbcluster-01)，然後選擇修改。

4. 在刪除保護下，清除啟用刪除保護核取方塊，然後選擇繼續。
5. 選擇立即套用，然後選擇修改叢集。

從先前的主要區域 () 在容錯回復程序期間叫用DR Orchestrator FAILBACK狀態機器us-east-1。

若要執行容錯回復，請執行下列動作：

1. 登入 主控台。
2. 將 區域變更為先前的主要區域 (us-east-1)。
3. 導覽至 服務，然後選擇 Step Functions。
4. 導覽至DR Orchestrator FAILBACK狀態機器。
5. 選擇開始執行，然後在 Input - optional區段中輸入下列 JSON 程式碼：

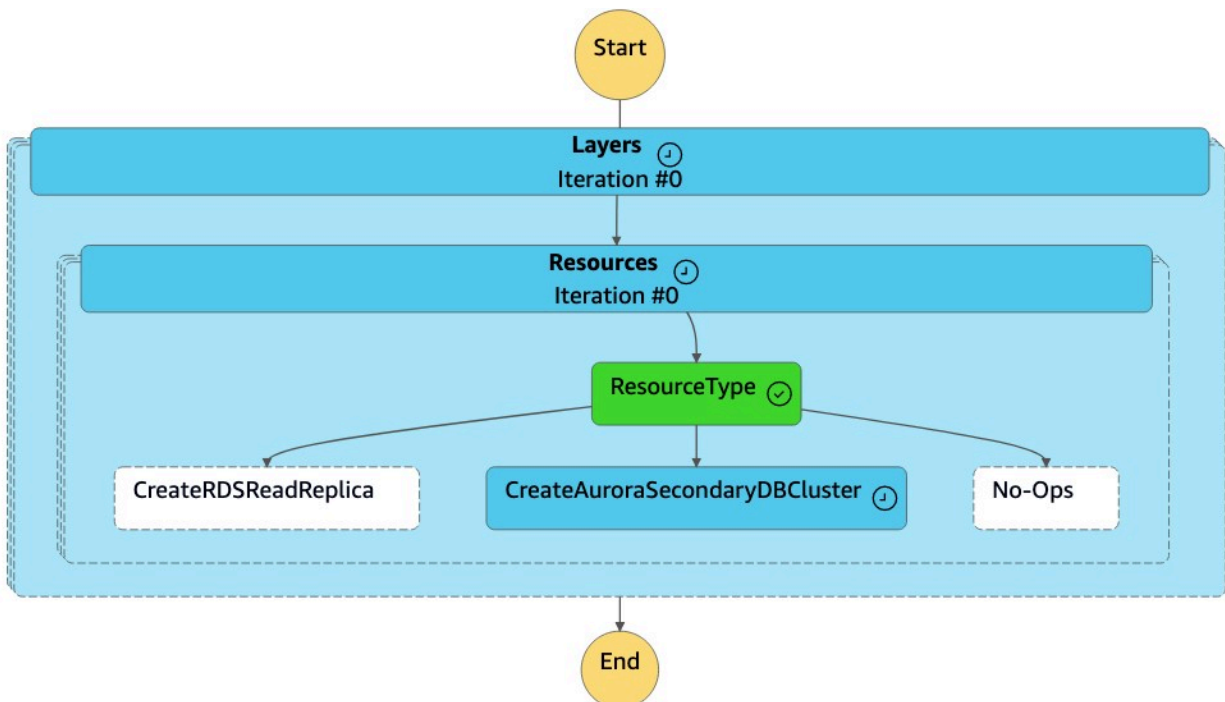
```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "CreateAuroraSecondaryDBCluster",
          "resourceName": "To create secondary Aurora MySQL Global Database Cluster",
          "parameters": {
            "GlobalClusterIdentifier": "!Import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

```

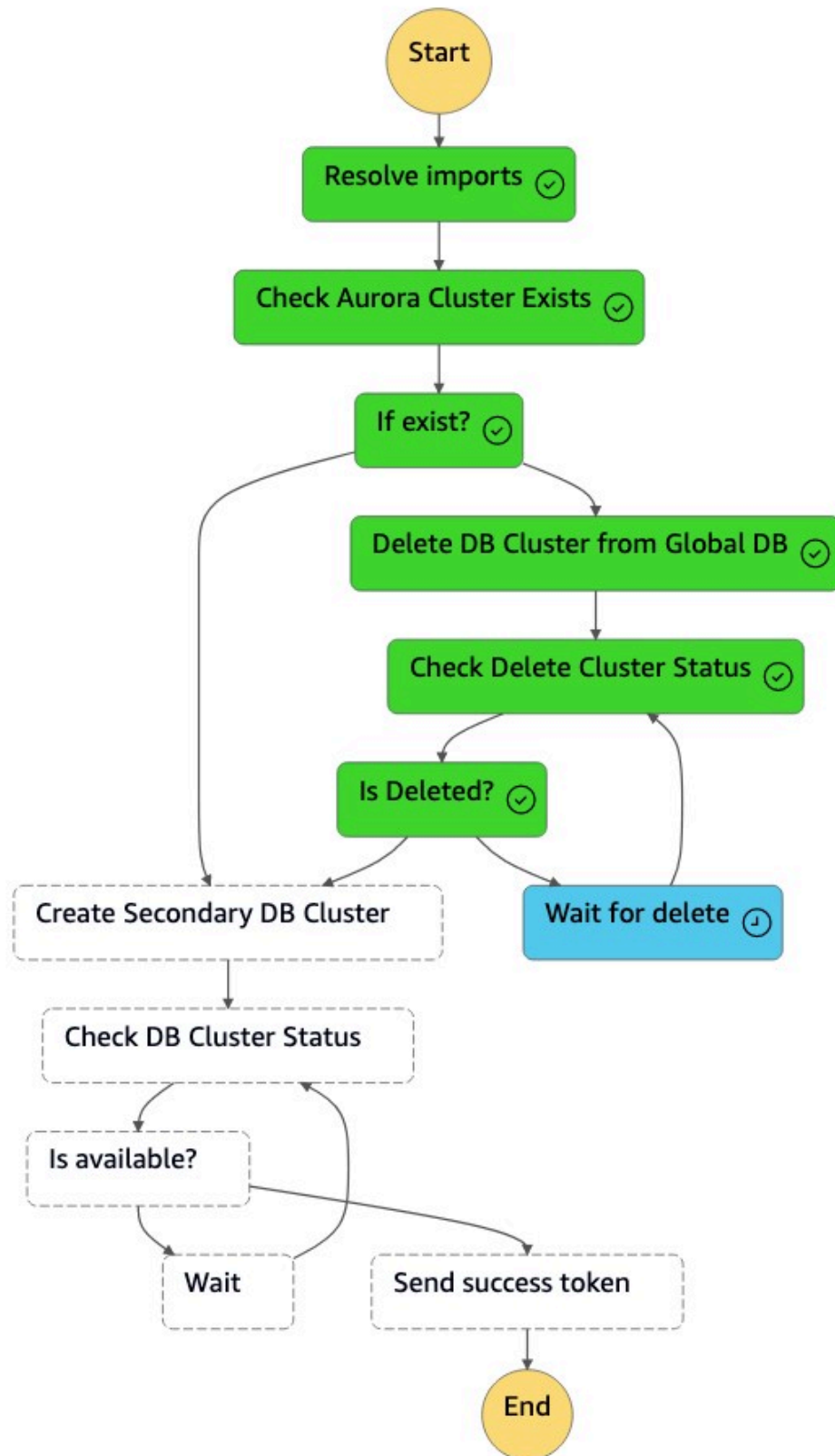
    "KmsKeyId": "!Import dr-globaldb-cluster-mysql-KmsKeyId",
    "SourceRegion": "!Import dr-globaldb-cluster-mysql-source-region",
    "ClusterRegion": "!Import dr-globaldb-cluster-mysql-cluster-region",
    "BackupRetentionPeriod": "7",
    "MonitoringInterval": "60",
    "StorageEncrypted": "True",
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

```

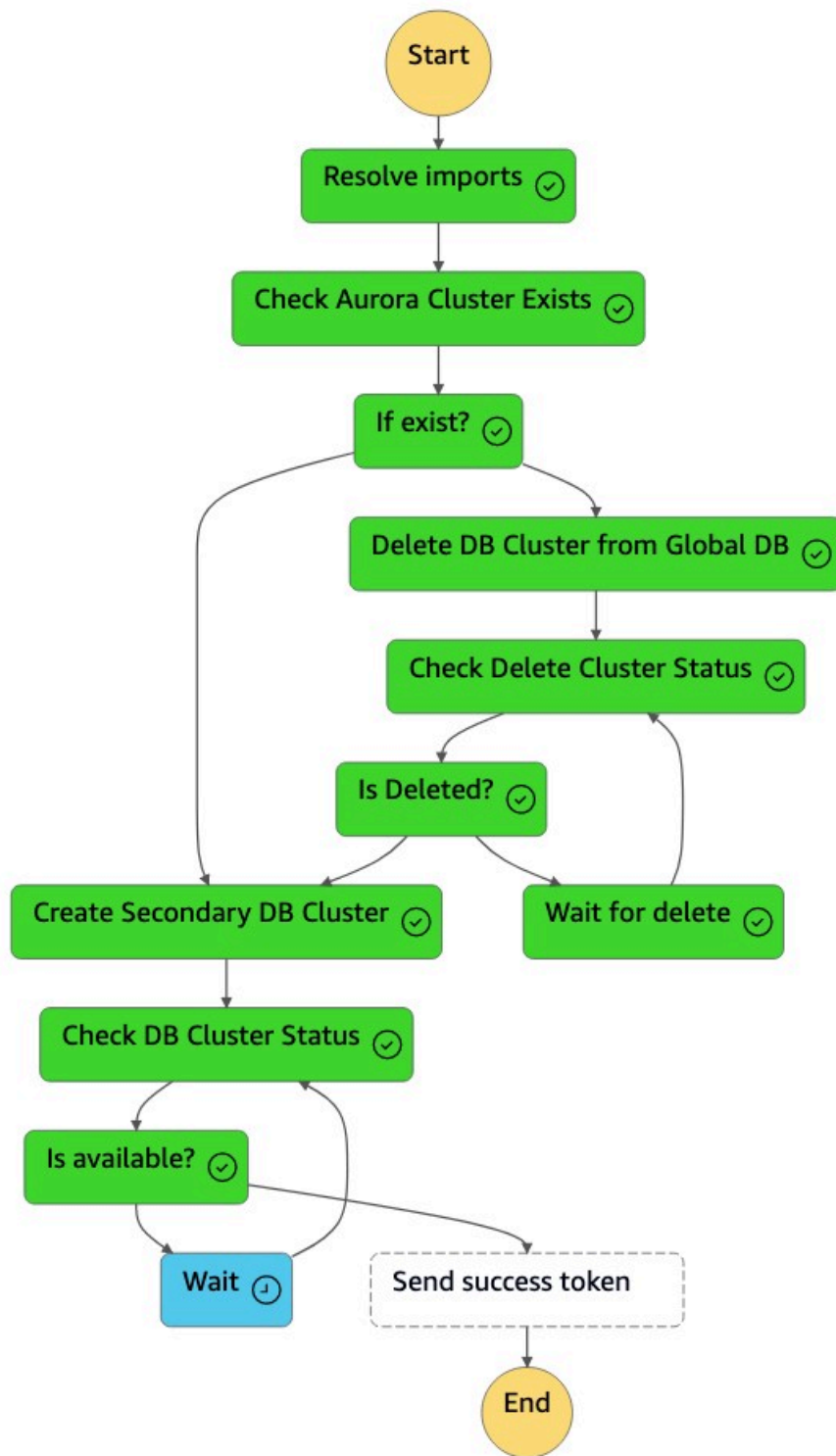
6. DR Orchestrator FAILBACK 狀態機器會將資源類型讀取為 `CreateAuroraSecondaryDBCluster`，並呼叫 `dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster` 狀態機器。



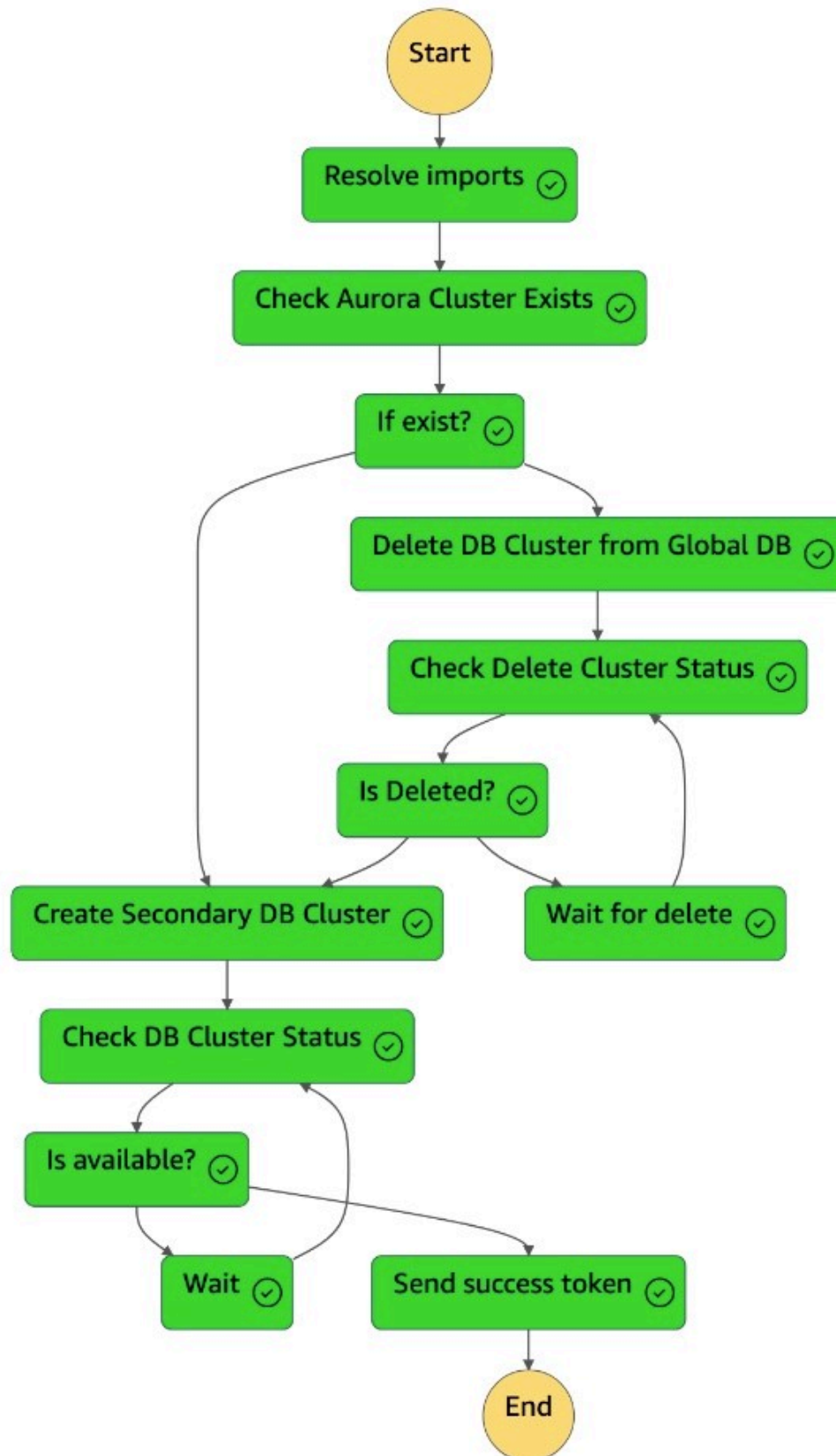
7. `dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster` 狀態機器會從先前的主要區域 (dbcluster-01) 刪除現有的叢集 (us-east-1)。



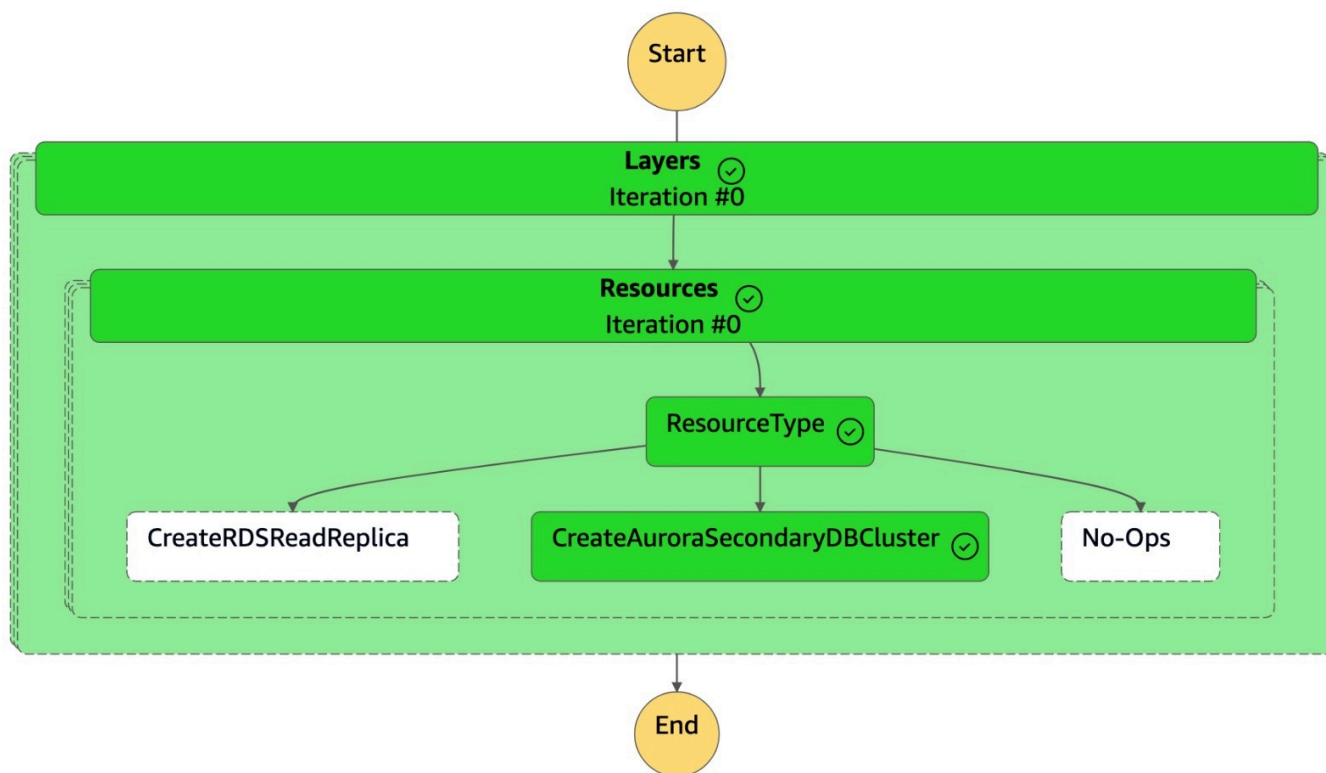
8. 刪除叢集 (dbcluster-01) 之後，狀態機器會與資料庫執行個體一起建立新的叢集 (dbcluster-01)，並加入 Aurora 全域資料庫做為次要叢集，以提供唯讀工作負載。



9. 次要叢集可用後，`dr-orchestrator-stepfunction-create-Aurora-Secondary-cluster`狀態機器即完成，並傳送成功字符回DR Orchestrator Failback狀態機器。



10dr-orchestrator-stepfunction-FAILBACK 狀態機器已完成。



11 您可以在 Amazon RDS 主控台上驗證 Aurora 全域資料庫。

如果您想要將主要資料庫叢集重新定位至 us-east-1，您可以遵循[切換](#)區段中提及的步驟。

# 部署解決方案

設定必要條件之後，請依照 [DR 協調程式架構模式](#) 中的指示進行。Amazon 和 Amazon RDS Aurora 包括說明 ElastiCache。若要測試解決方案，請檢閱先決條件和限制，並在上部署範例資料庫 AWS。

## 先決條件

- 主動[AWS 帳戶](#)部署此解決方案。
- 建立 [Amazon Simple Storage Service \(Amazon S3\)](#) 儲存貯體的許可，以及[AWS CloudFormation](#)範本中使用的資源。
- 在運行DR Orchestrator FAILBACK狀態機之前：
  - 手動[修改RDS執行個體](#)或[修改 Amazon Aurora 資料庫叢集](#)，然後停用DeletionProtection。
  - 如果您打算執行根本原因分析 (RCA)，請在刪除 Amazon RDS 執行個體或 Aurora 叢集之前拍攝[資料庫快照](#)。

## 限制

- 功能可用性和支援會因每個資料庫引擎的特定版本而有所不同 AWS 區域。如需跨區域複寫功能和區域可用性的詳細資訊，請參閱[跨區域僅供讀取](#)複本。
- 此解決方案已使用主要資料庫執行個體與一個僅供讀取複本進行測試。如果您想要使用一個以上的僅供讀取複本，請先徹底測試解決方案，然後再在生產環境中實作。
- 對於 AmazonRDS，該解決方案僅支持以下引擎：[我RDS的 Amazon SQL](#)，[Postgre RDS SQL 的 Amazon](#) 和 [Amazon RDS 的 MariaDB](#)。

## 部署範例資料庫 AWS

若要在上展示適用於資料庫的跨區域 DR 解決方案 AWS，您可以[RDS為我的SQL](#)、[Amazon Aurora 我的SQL相容版本](#)和 [Amazon ElastiCache \(RedisOSS\)](#)- 全球資料存放區部署亞馬遜。使用 [aws-cross-region-dr-數據庫 GitHub 存庫App-StackREADME文件中的](#)說明。在此範例中，主要「區域」為us-east-1，次要「區域」為us-west-2。

應用程式堆疊使用 [AWS CloudFormation [輸出](#)] 區段來匯出 DR Orchestrator Framework 所使用的 AWS 參數，以自動化資料庫的容錯移轉和容錯回復。

## 常見問答集

尋找有關容錯移轉和容錯回復的 DR Orchestrator 架構問題的答案。

### 使用此方法可以達到什麼 RPO 和 RTO ？

如需有關復原時間目標 (RTO) 和復原點目標 (RPO) 的資訊，請參閱 [資料庫的策略災難復原策略 AWS](#)。

### 是否必須使用 AWS CloudFormation 匯出變數？

否，您可以直接以 JSON 格式傳遞 Amazon Aurora 全域資料庫或 Amazon RDS 資料庫執行個體的值（例如：- "RDSInstanceIdentifier": "rds-mysql-instance"）。

### 我可以使用 DR Orchestrator FAILOVER 工作流程來容錯移轉多個 AWS 資料庫嗎？

是，您可以在輸入參數檔案中傳遞多個資源，以容錯移轉多個 AWS 資料庫。下列程式碼範例顯示平行容錯移轉 Amazon RDS for MySQL 僅供讀取複本和 Amazon ElastiCache (Redis OSS) 全域資料存放區：

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

## 執行 Amazon RDS 的 DR Orchestrator FAILBACK 狀態機器時，如何避免 InvalidParameterCombination 錯誤？

錯誤的完整文字為：

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

若要避免錯誤，[請在執行狀態機器之前停用 來修改 RDS 執行個體](#)。DeletionProtection DR Orchestrator FAILBACK

## 後續步驟

您可以使用 DR Orchestrator 架構，在啟動容錯移轉或容錯回復活動之前新增核准程序。例如，在 DR 事件期間啟動容錯移轉時，您可以在其上新增狀態機器，DR Orchestrator Failover 以透過電子郵件傳送 [Amazon Simple Notification Service \(Amazon SNS\)](#) 通知。一旦核准獲得，容錯移轉活動就會開始。

應用程式容錯移轉也可與 DR Orchestrator 架構整合，以便將 DNS 入口和指向應用程式切換到資料庫執行個體或叢集的新端點。

您可以使用 [AWS X-Ray](#) 追蹤來取得計算 RTO 的容錯移轉持續時間。您可以在 X-Ray 頂部構建監視儀表板。

## 資源

- [AWS 區域 使用全域資料存放區跨 複寫](#) (Amazon ElastiCache (Redis OSS) 文件 )
- [在 Amazon Aurora 全球資料庫中使用容錯移轉](#) (Aurora 文件)
- [自動化 Amazon Aurora Global Database 端點管理的計劃和非計劃容錯移轉](#) ( 部落格文章 )
- [災難復原：使用 Terraform 的 Amazon Aurora 全域資料庫的 3 個容錯移轉案例 \( 第 2 部分 \)](#) ( 部落格文章 )
- [資料庫在 上的災難復原策略 AWS](#) (AWS 方案指引策略 )
- [使用 DR Orchestrator Framework \( 方案指引模式 \) 自動化跨區域容錯移轉和容錯回復](#) AWS

# 文件歷史記錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 [RSS 摘要](#)。

| 變更                   | 描述 | 日期        |
|----------------------|----|-----------|
| <a href="#">初次出版</a> | —  | 2024年5月3日 |

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。