使用 Amazon DynamoDB 全域資料表

AWS 規定指引

AWS 規定指引: 使用 Amazon DynamoDB 全域資料表

Table of Contents

使用 Amazon DynamoDB 全域資料表

Amazon Web Services (AWS)，Jason Hunter

2024 年三月 (文件歷史記錄)

全域資料表建立於 Amazon DynamoDB 全域佈局的基礎之上，提供您全受管、多區域以及多個作用中的資料庫，為大幅度擴展的全域應用程式提供快速且本機的讀取與寫入的效能。全域資料表會自動複寫您所選擇的 DynamoDB 表格。 AWS 區域因為全域資料表使用現有的 DynamoDB API，所以不需要變更應用程式。使用全域資料表沒有前期成本或承諾，您僅需為使用的資源付費。

本指南說明如何有效使用 DynamoDB 全域資料表。它提供有關全域資料表的重要事實、說明功能的主要使用案例、為您介紹應該考慮的三種不同寫入模型的分類法、逐步介紹您可能實作的四種主要請求路由選項、討論疏散即時區域或離線區域的方法、說明如何考慮輸送容量規劃，以及提供部署全域資料表格時要考慮的核對清單。

本指南適用於更大的 AWS 多區域部署環境，如多區域基礎知識白皮書和影片的資料備援設計模AWS式所涵蓋。 AWS

內容

全域資料庫

關鍵事實

- 全域資料表有兩個版本：版本 2017.11.29 (舊版) (有時稱為 v1) 和版本 2019.11.21 (目前版本) (有時稱為 v2)。本指南僅著重於當前版本。
- DynamoDB (不含全域資料表) 是一項區域服務，這表示它具有高度可用性，並具有本質上能夠抵禦基礎架構故障，包括整個可用區域的故障。單一區域 DynamoDB 表格的設計可提供 99.99% 的可用性。如需詳細資訊，請參閱 DynamoDB 服務水準協議 (SLA)。
- DynamoDB 全域資料表會在兩個或多個區域之間複寫其資料。多區域 DynamoDB 表格的設計可提供 99.999% 的可用性。透過適當的規劃，全域資料表可協助建立能夠抵禦區域故障的架構。
- 全域資料表採用主動-主動式複寫模型。從 DynamoDB 的角度來看，每個區域中的資料表具有相同地位，可以接受讀取和寫入請求。收到寫入要求後，本機複本表格會在背景中將寫入作業複寫到其他參與的遠端區域。
- 單獨複製項目。在單一交易中更新的項目可能不會一起複寫。
- 來源區域中的每個資料表分割區都會與其他分割區 parallel 複寫其寫入作業。遠端區域內的寫入作業順序可能與來源區域內發生的寫入作業順序不相符。如需有關資料表分割區的詳細資訊，請參閱部落格文章擴展 DynamoDB：分割區、快捷鍵和熱分割會如何影響效能。
- 新寫入的項目通常會在一秒內傳播到所有複本列表。靠近區域的傳播速度往往更快。
- Amazon CloudWatch 為每個區域對提供一個`ReplicationLatency`指標。它是通過查看到達項目，將其到達時間與初始寫入時間進行比較，並計算平均值來計算。計時會儲存CloudWatch在來源區域內。檢視平均和最大計時對於判斷平均和最差情況的複寫延遲非常有用。此延遲沒有 SLA。
- 如果個別項目在兩個不同的區域中大約同時更新 (在此`ReplicationLatency`視窗內)，而第二個寫入作業在複寫第一個寫入作業之前發生，則可能會發生寫入衝突。全域資料表會根據寫入作業的時間戳記，使用最後一個寫入器 wins 機制來解決這類衝突。第一個操作「失去」到第二個操作。這些衝突不會記錄在CloudWatch或中AWS CloudTrail。
- 每個項目都有一個作為私有系統屬性保留的最後寫入時間戳記。最後一個作家 wins 方法是通過使用條件寫操作來實現的，該操作要求傳入項目的時間戳大於現有項目的時間戳。
- 全域表格會將所有項目複製到所有參與的區域。如果您想要有不同的複寫範圍，您可以建立多個全域資料表，並為每個表格指派不同的參與區域。
- 即使複本區域離線或`ReplicationLatency`成長，本機區域也會接受寫入作業。本機資料表會繼續嘗試將項目複製到遠端資料表，直到每個項目成功為止。

- 萬一區域完全離線，稍後重新上線時，所有擱置的輸出和輸入複製都會重試。不需要特殊動作即可使資料表恢復同步。最後一個作家 wins 機制可確保數據最終變得一致。
- 您可以隨時將新區域新增至 DynamoDB 資料表。DynamoDB 會處理初始同步處理和進行中的複寫作業。您也可以刪除區域（甚至是原始區域），這將刪除該區域中的本地表。
- DynamoDB 沒有全域端點。所有要求都會傳送至區域端點，該端點會存取該區域本機的全域資料表執行個體。
- 對 DynamoDB 的呼叫不應跨區域進行。最佳做法是針對一個區域的應用程式直接存取其區域的本機 DynamoDB 端點。如果在區域內 (在 DynamoDB 層或周圍堆疊中) 偵測到問題，則應將使用者流量路由到託管在不同區域的不同應用程式端點。全域資料表可確保每個區域中的應用程式都能存取相同的資料。

使用案例

全域資料表提供下列常見優點：

- 較低延遲的讀取作業。您可以將資料副本放置在靠近一般使用者的位置，以減少讀取作業期間的網路延遲。資料會保持與`ReplicationLatency`值一樣新鮮。
- 低延遲的寫入作業。最終用戶可以寫入附近的區域，以減少網絡延遲和完成寫入操作的時間。寫入流量必須小心路由，以確保沒有衝突。路由技術將在後面的章節中討論。
- 提高彈性和災難復原。如果某個區域的效能降低或完全中斷，您可以撤除該區域 (移除移至該區域的部分或所有要求)，並符合以秒為單位測量的復原點目標 (RPO) 和復原時間目標 (RTO)。使用全域資料表也會將每月正常執行時間百分比的 DynamoDB SLA 從 99.99% 提高到 99.999%。
- 無縫區域遷移。您可以新增區域，然後刪除舊區域，將部署從一個區域移轉到另一個區域，而不會在資料層停機。

例如，富達投資在「re: Invent 2022」上介紹了如何將 DynamoDB 全域表用於其訂單管理系統。他們的目標是以內部部署處理無法達到的規模實現可靠的低延遲處理，同時還能保持對可用區域和區域故障的彈性。

全域資料表的寫入模式

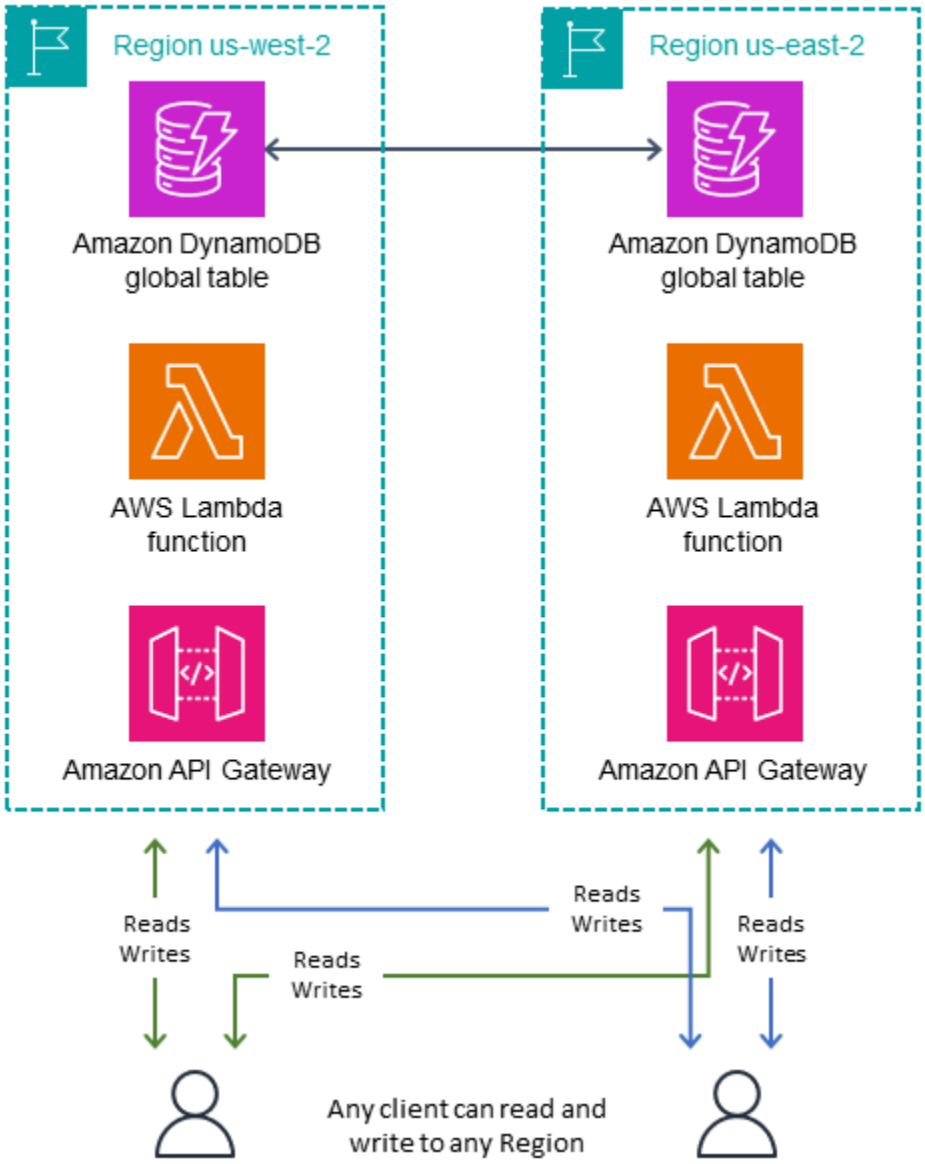
全域表在資料表永遠處於主動-主動資料表層級。不過，您可能想要透過控制路由，將它們用作主動-被動的方式。例如，您可能決定將寫入請求路由到單一區域，以避免潛在的寫入衝突。

有三個主要的託管寫入模式，如接下來三個部分所述。您應該考慮哪種寫入模式適合您的使用案例。此選項會影響您的路由請求、疏散區域以及處理災難復原的方式。後面章節中的指導取決於應用程式的寫入模式。

主題

寫入任何區域模式 (無優先層級)

寫入任何區域寫入模式是完全主動-主動的，並且不會對可能發生寫入操作的位置施加限制。任何地區都可以隨時接受寫入請求。這是最簡單的模式；不過，它只能用於某些類型的應用程式。當所有寫操作都是冪等的時候，這是合適的。冪等意味著它們可以安全地重複，以便跨區域的並發或重複寫入操作不會發生衝突，例如，當用戶更新其聯繫人數據時。它也適用於僅附加數據集，其中所有寫操作都是確定性主鍵下的唯一插入，這是冪等的特殊情況。最後，此模式適用於可接受衝突寫入操作的風險。



寫入任何區域模式是最直覺式的實作架構。因為任何區域都可以隨時成為寫入目標，路由會更加容易。容錯移轉比較容易，因為任何最近的寫入作業都可以重新顯示到任何次要區域。盡可能之下，您應該以此為寫入模式進行設計。

例如，數個影片串流服務會使用全域資料表來追蹤書籤、評論、觀看狀態旗標等。這些部署可以使用寫入任何區域模式，只要它們確保每個寫入作業都是冪等的。如果每次更新 (例如設定新的最新時間碼、指派新的審核或設定新的監看狀態) 都會直接指派使用者的新狀態，而項目的下一個正確值並不取決於其目前的值。如果偶然的情況下，用戶的寫入請求被路由到不同的區域，最後一個寫操作將持續，並且全局狀態將根據最後的分配結算。此模式下的讀取作業最終會變得一致，因為最新`ReplicationLatency`值會延遲。

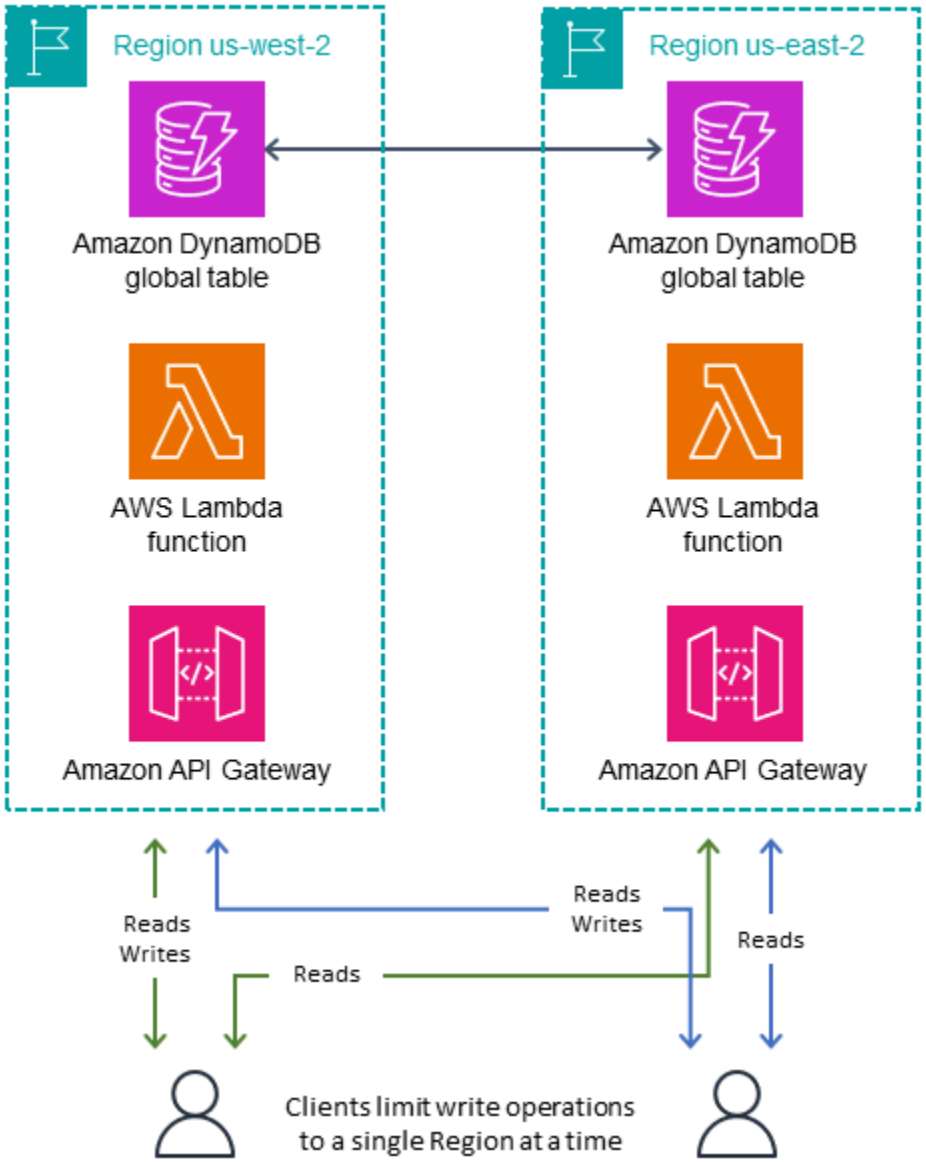
在另一個範例中，一家金融服務公司使用全域資料表作為系統的一部分來維護每個客戶使用借記卡購買的動作記錄，以計算該客戶的現金回饋獎勵。新的交易會從世界各地進行串流並前往多個區域。該公司能夠在仔細重新設計的情況下使用寫入任何區域模式。初始設計草圖為每位客戶保留一個RunningBalance項目。客戶動作會使用不是冪等的運ADD算式更新餘額 (因為新的正確值取決於目前值)，而且如果在不同區域中大約同時有兩個相同餘額的寫入作業，則餘額不同步。重新設計使用事件串流，其運作方式類似於具有僅附加工作流程的分類帳。每個客戶動作都會將新項目新增到為該客戶維護的項目收集器。(項目集合是共用主索引鍵但具有不同排序索引鍵的項目集合。) 每個寫入作業都是冪等插入，使用客戶 ID 做為分割索引鍵，而交易 ID 做為排序索引鍵。這種設計使得平衡的計算更加困難，因為它需Query要拉動項目，然後跟一些客戶端數學，但它使所有寫操作都是冪等的，並實現了路由和故障轉移的顯著簡化。本主題後續的「」。

第三個範例是提供線上廣告刊登位置服務的公司。該公司決定為了實現寫入任何區域模式的設計簡化，可以接受低資料遺失風險。當他們投放廣告時，他們只需幾毫秒就能擷取足夠的中繼資料來決定要顯示哪些廣告，然後記錄廣告曝光次數，這樣他們就不會很快重複相同的廣告。他們使用全域資料表來取得全球使用者的低延遲讀取作業，以及低延遲寫入作業。它們會在單一項目中記錄使用者的所有廣告曝光次數，表示為不斷增長的清單。他們使用一個項目而不是附加到項目集合中，因此可以在每次寫入操作中刪除較舊的廣告曝光，而無需支付刪除操作費用。這項寫入作業不是冪等的；如果同一位使用者看到在多個區域放送的廣告大約在同一時間，一個寫入作業的廣告曝光可能會覆寫另一個區域。風險是用戶可能會偶爾看到一次重複的廣告。他們認為這是可以接受的。

寫入單一區域模式 (單一優先層級)

寫入至一個區域寫入模式為主動-被動模式，並將所有資料表寫入作業路由至單一主動區域。(DynamoDB 沒有單一作用中區域的概念；DynamoDB 外部的圖層負責管理這個功能。) 寫入一個區域模式可確保寫入作業一次僅流向一個區域，以避免寫入衝突。當您想要使用條件運算式或交易時，此寫入模式會有所幫助。除非您知道自己正在針對最新資料採取行動，否則這些運算式是不可能的，因此它們需要將所有寫入請求傳送至具有最新資料的單一區域。

最終一致的讀取作業可以移至任何複本區域，以達到較低的延遲。強度一致的讀取作業必須移至單一主要區域。



有時候需要更改活動區域以應對區域故障，如後面所討論的那樣。有些使用者會定期變更目前作用中的[區域]，例如實作follow-the-sun部署。這會將活動區域放在活動最多的地理位置附近（通常是白天的位置，因此名稱），從而導致最低的延遲讀取和寫入操作。它還具有每天調用變更區域的代碼的副作用，並確保它在任何災難恢復之前都經過充分的測試。

被動區域可能會保留 DynamoDB 周圍的縮減規模基礎架構，該基礎架構只有在成為作用中區域時才會建立。本指南不涵蓋指示燈和暖待機設計。如需詳細資訊，您可以閱讀部落格文章的災難復原 (DR) 架構AWS，第三部分：指示燈和暖待命。

當您使用全域資料表進行低延遲、全域分散式讀取作業時，使用寫入一個區域模式很有效。一個例子是一家大型社交媒體公司，需要在世界各地的每個區域都具有相同的參考數據。他們不經常更新數據，但

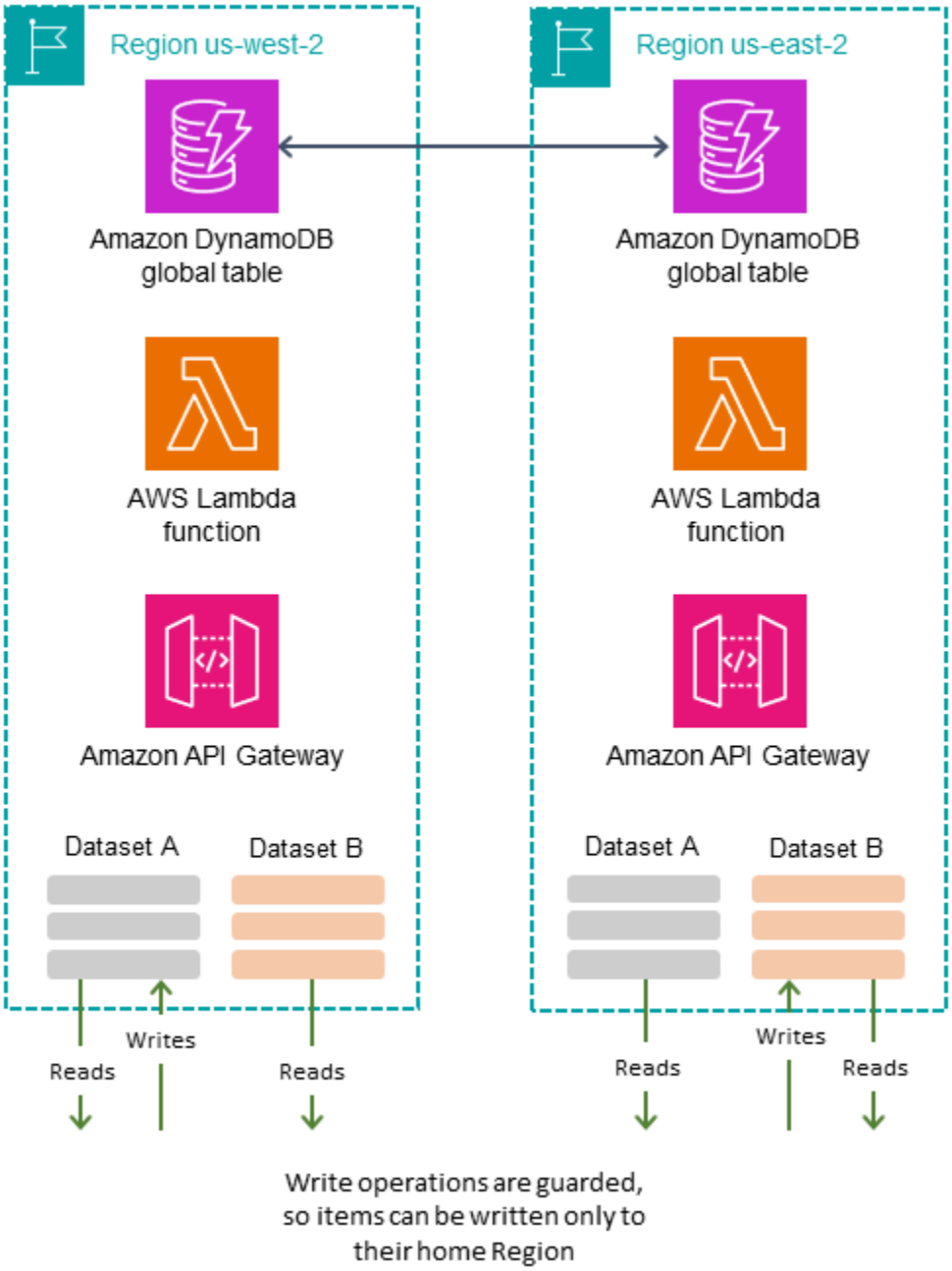
是當他們這樣做時，他們只寫入一個區域，以避免任何潛在的寫入衝突。讀取操作始終允許從任何區域。

作為另一個例子，考慮前面討論的金融服務公司實施了每日現金回饋計算。他們使用寫入任何區域模式來計算餘額，但寫入一個區域模式來跟踪現金返還款項。如果他們想為每花費 10 美元獎勵一分錢，則必須Query為前一天的所有交易，計算總支出，將現金返還決定寫入新表格，刪除查詢的項目集以將其標記為已消耗，並將其替換為單個項目，該項目存儲應進入第二天計算的任何剩餘部分。這項工作需要事務，因此在寫入一個 Region 模式時效果更好。只要工作負載沒有重疊的機會，即使在同一個資料表上，應用程式也可以混合寫入模式。

寫入您的區域模式 (混合優先層級)

寫入您的區域寫入模式會將不同的資料子集指派給不同的本位目錄區域，並且只允許透過項目的本地區域進行寫入作業。此模式是主動-被動，但根據項目分配活動區域。每個區域都是其本身非重疊資料集的主要資料集，而且必須保護寫入作業以確保適當的位置。

此模式類似於寫入一個區域，不同之處在於它允許更低延遲的寫入操作，因為與每個用戶關聯的數據可以放置在距離該用戶更接近的網絡中。它也會在區域之間更均勻地分散周圍的基礎結構，並且在容錯移轉案例期間建置基礎結構所需的工作較少，因為所有區域都有一部分的基礎結構已經在使用中。



您可以透過下列幾種方式決定物品的居住地區域：

- 內在：數據的某些方面（例如特殊屬性或嵌入其分區鍵中的值）使其主區域清晰。此技術在部落格文章中說明使用區域固定功能為 Amazon DynamoDB 全域表中的項目設定主區域。
- 已協商：每個資料集的本位目錄區域會以某種外部方式進行交涉，例如與維護指派的個別全域服務進行交涉。分配可能有一個有限的持續時間，之後它需要重新協商。

- 資料表導向：您不需要建立單一複製的全域表格，而是建立與複寫區域相同數量的全域表格。每個資料表的名稱都表示其主區域。標準操作中，所有資料都會寫入主區域，而其他區域則保留唯讀副本。在容錯移轉期間，另一個區域會暫時採用該資料表的寫入職責。

例如，假設您正在為一家遊戲公司工作。您需要為全球所有遊戲玩家提供低延遲的讀取和寫入操作。您可以將每位玩家指派到離他們最近的地區。該區域接受所有讀取和寫入操作，以確保強大的read-after-write一致性。但是，當玩家旅行或其所在地區遭受中斷時，其資料的完整副本可在替代區域取得，而玩家可以指派至不同的居住地區。

另一個例子，假設您正在視訊會議公司工作。每個電話會議的中繼資料都會指派給特定區域。來電者可以使用最接近他們的區域來實現最低的延遲。如果發生區域中斷，使用全域資料表可讓您快速復原，因為系統可將呼叫的處理移至已存在複寫資料副本的其他區域。

全域表的路由策略

也許，全域資料表部署中最複雜的部分是管理請求路由。請求必須首先從終端使用者以某種方式選擇和路由到區域。請求遇到該區域中的某些服務堆疊，包括一個運算層，該運算層可能由 AWS Lambda 函數、容器或 Amazon 彈性運算雲端 (AmazonEC2) 節點支援的負載平衡器組成，以及可能包括另一個資料庫在內的其他服務。該計算層會與 DynamoDB 通訊。它應該通過使用該區域的本地端點來做到這一點。全域資料表中的資料會複寫到所有其他參與的區域，而且每個區域在其 DynamoDB 資料表周圍都有類似的服務堆疊。

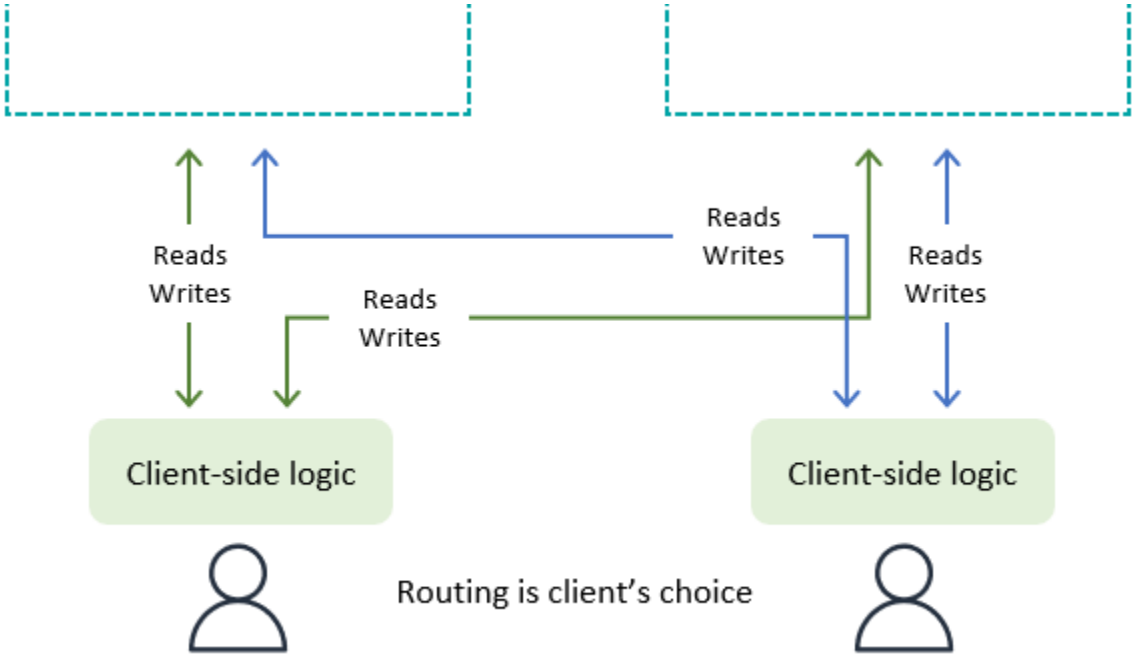
全域資料表會為不同區域中的每個堆疊提供相同資料的本機複本。如果本機 DynamoDB 資料表發生問題，您可以考慮為單一區域中的單一堆疊進行設計，並預期會對次要區域的 DynamoDB 端點進行遠端呼叫。這不是最佳做法。跨區域相關的延遲可能比本機存取高 100 倍。在本地執行一 back-and-forth 系列 5 個請求可能需要幾毫秒，但穿越全球時可能需要幾秒鐘的時間。建議最好將終端使用者路由到另一個區域進行處理。為了確保備援，您需要跨多個區域進行複寫：複寫運算層以及資料層。

有許多技術可將最終用戶請求路由到區域進行處理。正確的選擇取決於您的寫入模式和容錯移轉考量。本節討論四個選項：用戶端導向、運算層、Amazon Route 53 和. AWS Global Accelerator

主題

用戶端驅動的請求路由

透過用戶端導向的請求路由，最終使用者用戶端 (應用程式 JavaScript、網頁或其他用戶端) 會追蹤有效的應用程式端點 (例如，Amazon API Gateway 端點而非常值 DynamoDB 端點)，並使用其自己的內嵌邏輯來選擇要與之通訊的區域。它可能會根據隨機選擇、觀察到的最低延遲、觀察到的最高頻寬測量值或本機執行的健康狀態檢查來選擇。



作為一個優勢，客戶導向的請求路由可以適應諸如現實世界公共互聯網流量條件之類的事情，以在發現任何性能下降時切換區域。用戶端必須瞭解所有可用端點，但啟動新的區域端點並不常見。

透過寫入任何區域模式，用戶端可以單方面選取其偏好的端點。如果對某個區域的存取受損，用戶端可以路由到另一個端點。

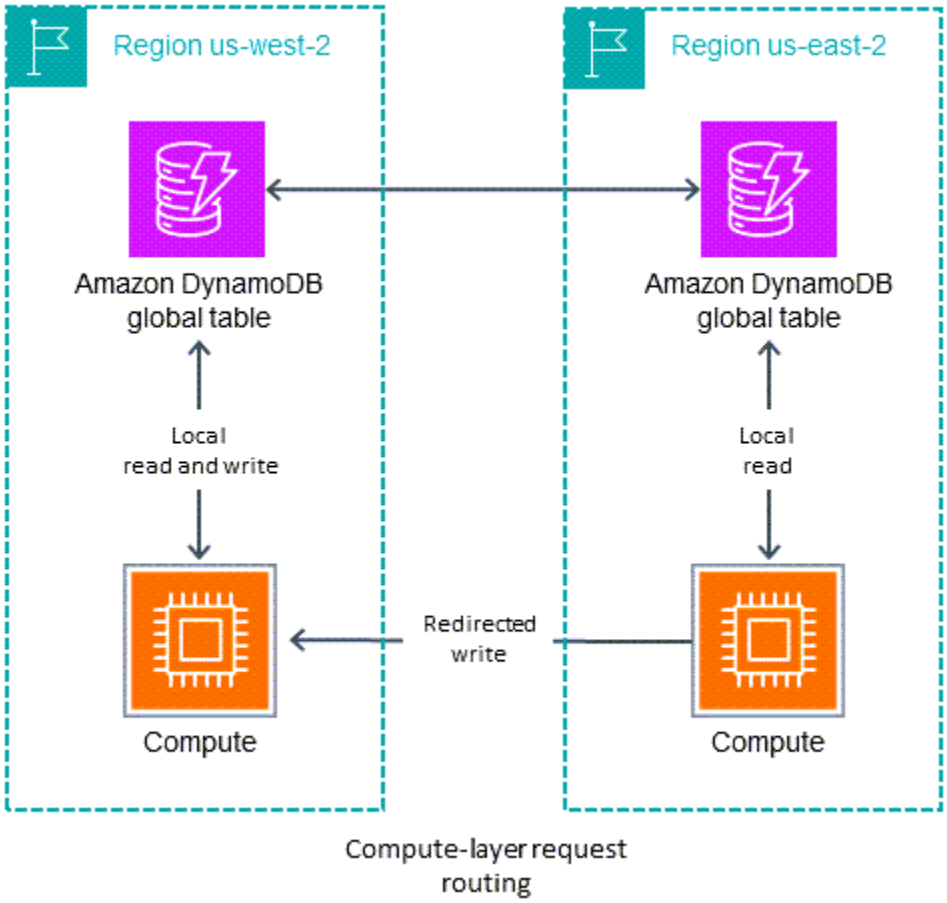
使用寫入到一個區域模式，客戶端需要一種機制，將其寫入請求路由到當前活動的 Region。這可能是一種基本機制，例如經驗測試目前哪個 Region 正在接受寫入請求（注意任何寫入拒絕並退回到替代請求）。或者，它可能是一種複雜的機制，例如使用全域協調器查詢目前的應用程式狀態 (可能建立在 Amazon Application Recovery Controller (ARC) (ARC) 路由控制上，該控制項提供五個區域、法定值驅動的系統來維護全域狀態以滿足此類需求）。客戶可以決定讀取請求是否可以轉到任何區域以獲得最終一致性，還是必須路由到活動區域以獲得強大的一致性。

透過寫入您的區域模式，用戶端必須判斷其使用之資料集的主區域。例如，如果用戶端對應於一個使用者帳戶，且每個使用者帳戶都連結至某個區域，則用戶端可以要求適當的端點指派，以便從全域登入系統搭配其認證使用。

例如，協助使用者透過網路管理其商業財務的金融服務公司，會使用全域資料表並寫入您的區域模式。每個使用者都必須登入中央服務。該服務會傳回認證以及這些認證將運作之區域的端點。傳回的區域是根據使用者資料集目前所在位置而定。憑證僅在短時間內有效。之後，網頁會自動協商新登錄，這提供了可能將用戶活動重定向到新區域的機會。

運算層請求路由

透過運算層要求路由，在計算層中執行的程式碼會決定是在本機處理要求，還是將要求傳遞給在另一個區域中執行的本身副本。當您使用寫入至一個區域模式時，運算層可能會偵測到它不是使用中的區域，並允許本機讀取作業，同時將所有寫入作業轉送至另一個區域。此計算層程式碼必須瞭解資料拓撲和路由規則，並根據指定哪些區域為哪些資料作用中的最新設定，可靠地強制執行這些規則。區域內的外部軟體堆疊不需要知道微服務如何路由讀取和寫入請求。在穩健的設計中，接收區域會驗證其是否為寫入操作的目前主要項目。如果不是，則會產生錯誤，指出需要全域狀態需要修改。如果主要區域正在變更，則接收區域也可能會緩衝寫入操作一段時間。在所有情況下，區域中的運算堆疊只會寫入其本機 DynamoDB 端點，但運算堆疊可能會彼此通訊。

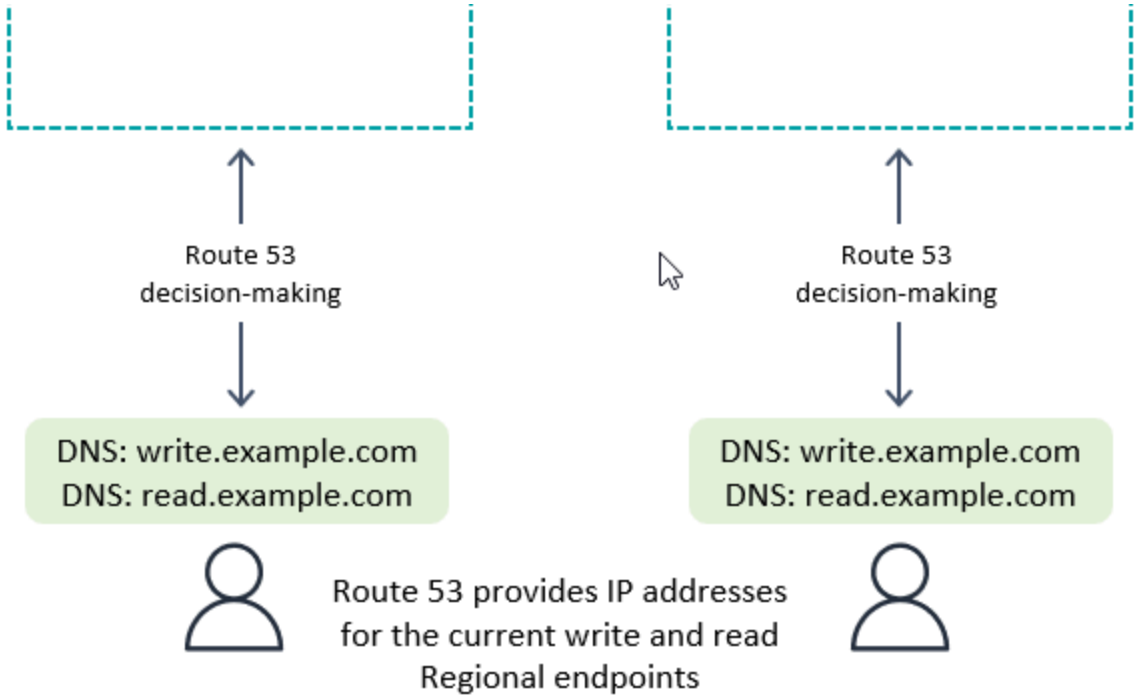


先鋒集團使用名為「全球協調和狀態工具」（GOaST）的系統和一個名為「全球多區域庫」（GMRlib）的庫來進行此路由流程，如 Re：Invent 2022 中所述。它們使用 follow-the-sun 單一主要模型。GOaST維持全域狀態，類似於前一節中討論的ARC路由控制項。它使用全域資料表來追蹤哪個區域是主要區域，以及排定下一個主要交換器的時間。所有讀取和寫入操作都經過GMRlib，與之協調GOaST。GMRlib允許在本機以低延遲執行讀取作業。對於寫操作，GMRlib檢查本地區域是否是當前的主要區域。如果是，則寫入操作會直接完成。如果不是，請GMRlib將寫入任務轉發到主要區域

GMRlib中的。接收程式庫會確認它也會將自己視為主要區域，如果不是，則會引發錯誤，表示全域狀態的傳播延遲。此方法不直接寫入遠端 DynamoDB 端點，進而提供驗證優勢。

Route 53 請求路由

Amazon 路線 53 是一種域名服務（DNS）技術。使用 Route 53 時，用戶端會透過查詢已知的DNS網域名稱來要求其端點，Route 53 會傳回與其判定最合適的區域端點對應的 IP 位址。Route 53 有很長的路由政策清單，用於確定適當的區域。它也可以執行容錯移轉路由，將流量路由傳送出健康狀態檢查失敗的區域。



使用寫入任何區域模式，或者與後端的計算層請求路由結合使用，Route 53 可以根據任何複雜的內部規則（例如在最近的網絡或地理位置接近或任何其他選擇中選擇區域）完全自由返回 Region。

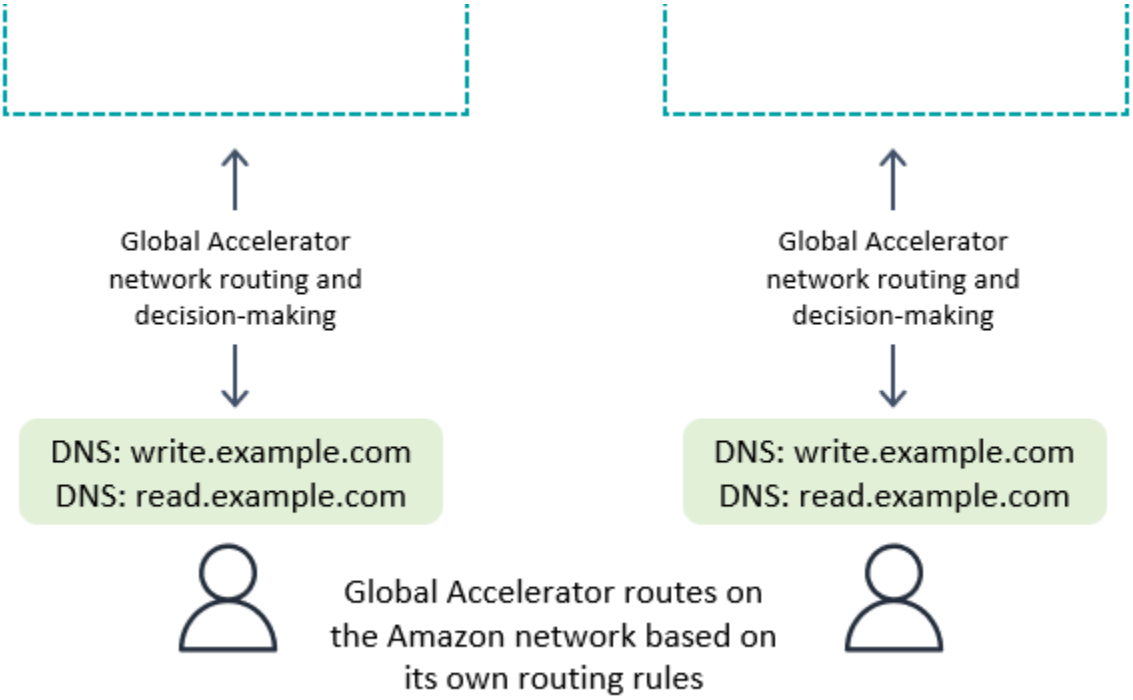
使用寫入一個區域模式，您可以將 Route 53 配置為返回當前活動的區域（使用ARC）。如果用戶端想要連線到被動區域 (例如，讀取作業)，它可以查詢不同的DNS名稱。

> **Note**
>
> 用戶端會快取 Route 53 回應中的 IP 位址，一段時間 (該時間為網域名稱上的存留時間 (TTL) 設定。TTL延長所有用戶端辨識新端點的復原時間目標 (RTO)。60 秒是容錯移轉使用的典型值。並非所有軟體都能完全遵守到DNSTTL期，而且可能會有多個層級的DNS快取，例如在作業系統、虛擬機器和應用程式上。

寫入您的區域模式時，最好避免使用 Route 53，除非您還使用計算層請求路由。

Global Accelerator 請求路由

使AWS Global Accelerator用客戶端查找 Route 53 中的知名域名。不過，用戶端不會取回對應至區域端點的 IP 位址，而是取回路由至最近 AWS 邊緣位置的任何傳播靜態 IP 位址。從該節點開始，所有流量都會在私人 AWS 網路上路由傳送至 Global Accelerator 內維護的路由規則所選擇的區域中的某些端點 (網路負載平衡器、應用程式負載平衡器、EC2執行個體或彈性 IP 位址)。與使用 Route 53 規則為基礎的路由相比，Global Accelerator 請求路由的延遲較低，因為它可以減少公用網際網路上的流量。此外，由於全域加速器不依賴DNSTTL到期時間來變更路由規則，因此可以更快速地調整路由。



通過寫入任何區域模式，或者如果與後端的計算層請求路由結合使用，Global Accelerator 可以無縫運行。用戶端會連線到最近的邊緣位置，而且不必擔心哪個區域會接收要求。

在「寫入至一個區域」模式下，「全域加速器」路由規則必須傳送請求至目前作用中的「區域 您可以使用運作狀態檢查，以人為方式報告任何區域的故障，而這些區域並未被您的全域系統視為使用中區域。如同DNS，如果請求可以來自任何區DNS域，則可以使用替代網域名稱來路由讀取請求。

寫入您的區域模式時，最好避免使用全域加速器，除非您同時使用運算層要求路由。

全域資料表撤離程序

撤離區域是將活動（通常是寫入活動，可能讀取活動）遠離該區域的過程。

疏散即時區域

您可能基於多種原因決定撤離即時區域：作為一般業務活動的一部分 (例如，如果您使用的是寫入一個 Region 模式)、因為商業決定變更目前作用中的區域、以回應 DynamoDB 外部軟體堆疊的故障，或是因為您遇到一般問題，例如區域內一般延遲高。follow-the-sun

通過寫入任何區域模式，疏散即時區域非常簡單。您可以使用任何路由系統將流量路由到替代區域，並讓已撤離區域中已發生的寫入作業如常複寫。

透過寫入至一個區域並寫入您的區域模式，您必須確定在新的使用中區域中開始寫入作業之前，已完整記錄、串流處理和全域傳播至作用中區域的所有寫入作業，以確保 future 的寫入作業會根據最新版本的資料處理。

假設區域 A 處於作用中狀態，而區域 B 是被動的 (無論是針對全域資料表或區域 A 中的項目來說)。執行疏散的典型機制是暫停寫入 A、等待充分的時間讓這些操作完全傳播到 B、更新架構堆疊以辨識 B 為使用中，然後繼續寫入操作至 B。沒有指標能夠保證區域 A 的資料已完全複製到區域 B。如果區域 A 狀況良好，請暫停寫入操作至區域 A，然後等待 10 倍 `ReplicationLatency` 最近指標的最大值，判斷複製是否完成。如果區域 A 狀況不佳並顯示延遲增加的其他區域，則您可以選擇較大的倍數作為等待時間。

疏散離線區域

有一個特殊情況需要考慮：如果 A 區域在沒有通知的情況下完全離線怎麼辦？ 這是極不可能的，但仍應考慮。如果發生這種情況，區域 A 中尚未傳輸的任何寫入操作都會在區域 A 重新上線後保留和傳播。寫操作不會遺失，但它們的傳播會無限期延遲。

在這種情況下如何進行應用程式的決策。對於業務連續性，寫入操作可能需要繼續進行新的主要區域 B。不過，如果區域 B 中的某個項目在區域 A 的寫入操作擱置傳播時收到更新，則會在最後一個寫入獲勝模型下抑制傳播。區域 B 中的任何更新都可能會抑制傳入的寫入請求。

透過寫入任何區域模式，區域 B 中的讀取和寫入作業可以繼續進行，相信區域 A 中的項目最終會傳播到區域 B，並識別遺失項目的可能性，直到區域 A 重新連線為止。如果可能的話，例如使用冪等寫入作業，您應該考慮重新播放最近的寫入流量 (例如，使用上游事件來源)，以填補任何可能遺失寫入作業的空白，並讓最後一個寫入器 wins 衝突解決方案抑制傳入寫入作業的最終傳播。

使用其他寫入模式，您必須考慮工作可以稍微out-of-date觀察世界的程度。區域 A 重新上線之前，將丟失一些持續時間較短的寫入操作 (如，ReplicationLatency 追蹤)。業務能夠繼續進行嗎？ 在某些使用案例中，能夠繼續進行，但在其他情況下，如果沒有額外的緩解機制，可能無法繼續進行。

例如，假設即使在區域完全中斷之後，您也必須維持可用的信用餘額而不會中斷。您可以將餘額分成兩個不同的項目，一個在區域 A 中，另一個在區域 B 中，每個項目都有一半的可用餘額。這將使用寫入您的區域模式。每個區域中處理的交易更新會針對餘額的本機複本寫入。如果區域 A 完全離線，工作仍然可以在區域 B 中繼續進行交易處理，並且寫入操作將限制為區域 B 中所保留的餘額部分，像這樣拆分餘額會導致複雜性，但即使在不確定的待處理寫入操作下，可以提供一個安全業務復原的範例。

作為另一個例子，假設您正在捕獲 Web 表單數據。您可以使用樂觀並行控制項 (OCC) 將版本指派給資料項目，並將最新版本內嵌到 Web 表單中做為隱藏欄位。每次提交時，只有當資料庫中的版本仍與建置表單的版本相符時，寫入操作才會成功。如果版本不相符，則可以根據資料庫中目前版本重新整理 (或仔細合併) Web 表單，並且使用者可以再次進行操作。OCC 模型通常可以防止其他用戶端覆寫和產生新版本的資料，但也可以在用戶端遇到較舊版本資料的容錯移轉期間提供幫助。假設您是使用時間戳記作為版本。表單首先是在 12:00 針對區域 A 建立，但 (容錯移轉之後) 嘗試寫入區域 B，並注意資料庫中的最新版本為 11:59。在此情況中，用戶端可以等候 12:00 版本傳播到區域 B，然後在該版本之上寫入，或是在 11:59 建置並建立新的 12:01 版本 (寫入後，會在區域 A 復原之後抑制傳入版本)。

第三個範例是，金融服務公司在 DynamoDB 資料庫中保存客戶帳戶及其財務交易的相關資料。如果區域 A 完全中斷，他們希望確保與其帳戶相關的任何寫入活動在區域 B 中可用，或者想要隔離已知的部分帳戶，直到區域 A 重新上線為止。他們沒有暫停所有業務，而是決定只對確定有未傳播交易的一小部分賬戶暫停業務。為了達成此目的，他們使用了第三個區域，我們將呼叫區域 C。在處理區域 A 中的任何寫入操作之前，他們在區域 C 中對那些暫緩操作 (例如，帳戶新的交易計數) 建立摘要彙總。此彙總足以讓區域 B 判斷其檢視是否完全為最新狀態。此動作會有效鎖定帳戶，從寫入區域 C 開始，直到區域 A 接受寫入操作且區域 B 收到為止。除非作為容錯移轉程序的一部分外，不會使用區域 C 中的資料，之後區域 B 可以和區域 C 交叉比對資料，以檢查其帳戶是否已過期。這些帳戶將被標記為隔離，直到區域 A 恢復將部分資料傳播到區域 B。如果區域 C 失敗，則可能會改用新的區域 D。區域 C 中的資料非常暫時，幾分鐘後，區域 D 就會有足夠的執行中寫入作業up-to-date記錄，以便完全有用。如果區域 B 當機，區域 A 可以繼續接受與區域 C 合作的寫入請求。這家公司願意接受更高的延遲寫入 (到兩個區域：C 和 A)，並且很高興擁有資料模型，可以摘要彙總帳戶狀態。

全域資料表的輸送容量規劃

將流量從一個區域移轉到另一個區域需要仔細考慮與容量有關的 DynamoDB 資料表設定。

以下是管理寫入容量的一些注意事項：

- 全域資料表必須處於隨需模式，或啟用自動擴展的佈建。
- 如果使用自動擴展進行佈建，則會跨區域複寫寫入設定 (最小、最大和目標使用率)。雖然自動擴展的設定是同步的，但實際佈建的寫入容量可能會在區域之間獨立浮動。
- 您可能會看到不同的佈建寫入容量的原因之一是使用時間 (TTL) 功能。當您TTL在 DynamoDB 中啟用時，您可以指定屬性名稱，其值表示項目的到期時間，單位為 Unix 紀元時間格式 (以秒為單位)。之後，DynamoDB 可刪除項目，而不會產生寫入成本。透過全域表格，您可以TTL在任何區域中進行設定，而且設定會自動複製到與全域表格相關聯的其他區域。當料號符合透過TTL規則刪除資格時，可在任何區域中完成該工作。執行刪除作業時不會耗用來源資料表上的寫入單位，但複本表格會取得該刪除作業的複寫寫入作業，並會產生複寫的寫入單位成本。
- 如果您使用 auto 擴展，請確定佈建的寫入容量上限設定足以處理所有寫入作業以及所有潛在的TTL刪除作業。自動擴展會根據每個區域的寫入耗用進行調整。隨需資料表沒有最大佈建的寫入容量設定，但是資料表層級最大寫入輸送量限制會指定隨需資料表允許的最大持續寫入容量。預設限制為 40,000，但可以調整。我們建議您將它設定得足夠高，以處理隨選資料表可能需要的所有TTL寫入作業 (包括寫入作業)。當您設定全域資料表時，所有參與區域的這個值必須相同。

以下是管理讀取容量的一些注意事項：

- 允許區域之間的讀取容量管理設定不同，因為需假設不同的區域可能具有獨立的讀取模式。當第一次將全域複本新增至資料表時，會傳播來源區域的容量。建立之後，您可以調整讀取容量設定，而這個設定不會傳輸到另一端。
- 使用 DynamoDB 自動擴展時，請確保佈建的最大讀取容量設定充足，以處理所有區域的所有讀取操作。在標準操作期間，讀取容量可能會分散到區域，但在容錯移轉期間，資料表應該能夠自動適應增加的讀取工作負載。隨需資料表沒有最大佈建的讀取容量設定，但是資料表層級最大讀取輸送量限制會指定隨需資料表允許的最大持續讀取容量。預設限制為 40,000，但可以調整。如果所有讀取操作都要路由到此單一區域，我們建議您將它設定充足，以處理資料表可能需要的所有讀取操作。
- 如果一個區域中的資料表通常不會收到讀取流量，但在容錯移轉後可能必須吸收大量的讀取流量，您可以提高資料表的佈建讀取容量，等待資料表完成更新，然後再縮減佈建資料表。您可以將資料表保留為佈建模式，也可以將其切換為隨需模式。這會預熱資料表，以接受更高層級的讀取流量。

ARC具有整備檢查，可用於確認 DynamoDB 區域具有類似的表格設定和帳戶配額，無論您是否使用 Route 53 路由傳送請求。這些整備程度檢查也可協助您調整帳戶層級配額，使其符合。

全域表的準備檢查清單

部署全域資料表時，請使用下列檢查清單來制定決策和執行任務。

- 決定有多少以及哪些區域應參與全域資料表。
- 判斷應用程式的寫入模式。
- 根據您的寫入模式規劃路由策略。
- 根據您的寫入模式和路由策略定義疏散計劃。
- 擷取每個區域的運作狀態、延遲和錯誤的指標。如需 DynamoDB 指標的清單，請參閱 AWS 部落格文章監控 Amazon DynamoDB 以瞭解營運感知。您還應該使用合成金絲雀（旨在檢測故障的人工請求）以及實時觀察客戶流量。並非所有問題都會出現在 DynamoDB 指標中。
- 為 `ReplicationLatency` 中任何持續增加設定警示。增加可能表示意外設定錯誤，而全域資料表在不同區域中有不同的寫入設定，這會導致複寫請求失敗並增加延遲。這也可能表示存在區域中斷。一個很好的例子是如果最近的平均值超過 180,000 毫秒，則發出警示。您也可能會注意`ReplicationLatency`下降到 0，這表示停止複寫。
- 為每個全域資料表指派充足的最大讀取和寫入設定值。
- 確定您將撤離區域的條件。如果決定涉及人為判斷，請記錄所有考量因素。這項工作應提前仔細完成，而不是在壓力下進行。
- 為每個動作維護執行手冊，作為當疏散區域時必須採取的措施。通常，全域資料表涉及的工作很少，但移動堆疊的其餘部分可能很複雜。

> **Note**
>
> 使用容錯移轉程序時，最佳做法是僅仰賴資料平面作業，而非控制平面作業，因為某些控制平面作業在區域故障期間可能會降級。如需詳細資訊，請參閱部 AWS 落格文章使用 Amazon DynamoDB 全域表建置彈性應用程式：第 4 部分。

- 定期測試執行手冊的各個方面，包括區域疏散。未經測試的執行手冊是不可靠的執行手冊。
- 請考慮使用AWS Resilience Hub來評估整個應用程式 (包括全域資料表) 的彈性。此服務透過其儀表板，提供應用程式產品組合復原狀態的全面檢視。
- 請考慮使用ARC整備程度檢查來評估應用程式目前的組態，並追蹤最佳實務的任何差異。
- 當您撰寫健康狀態檢查以搭配 Route 53 或全域加速器使用時，請進行一組涵蓋完整資料庫流程的呼叫。如果您將檢查限制為僅確認 DynamoDB 端點已啟動，則無法涵蓋許多失敗模式，例如 AWS

Identity and Access Management (IAM) 組態錯誤、程式碼部署問題、DynamoDB 外部堆疊中的故障、高於平均讀取或寫入延遲等等。

全域資料表常見問答集

此區段提供有關 DynamoDB 全域資料表的常見問答集解答。

全域資料表的定價為何？

- 傳統 DynamoDB 資料表的寫入作業是以適用於佈建資料表的寫入容量單位 (WCU) 或適用於隨需資料表的寫入請求單位 (WRU) 來定價。如果您寫入一個 5 KB 的項目，則會產生 5 個單位的費用。全域資料表的寫入是以複寫寫入容量單位 (rWCU，適用於佈建資料表) 或複寫寫入請求單位 (rWRU，適用於隨需資料表) 定價。
- rWCU 和 rWRU 包含管理複寫所需的串流媒體基礎設施成本。因此，它們的價格比 WCU 和 WRU 高出 50%。需支付跨區域資料傳輸費用。
- 在直接寫入項目或透過複寫寫入項目的每個區域，都會產生 RWCU 和 RWRu 費用。
- 寫入全域次要索引 (GSI) 會視為本機寫入作業，並使用常規寫入單位。
- rWCU 目前沒有可用的預留容量。對於 GSI 消耗寫入單元的資料表而言，購買 WCU 的預留容量可能仍然有所幫助。
- 當您將新區域加入全域資料表時，DynamoDB 會自動啟動新區域，並根據資料表的 GB 大小向您收費，就像是資料表還原一樣。此外，需支付跨區域資料傳輸費用。

全域資料表支援哪些區域？

全域資料表支援所有 AWS 區域。

GSI 如何處理全域資料表？

在全域資料表 (目前，版本 2019) 中，當您在某個區域中建立 GSI 時，它會在其他參與的區域中自動建立並自動回填。

如何停止全域資料表的複寫？

刪除複本資料表的方式，與刪除其他資料表相同。刪除全域資料表將停止複寫到該區域，並刪除保留在該區域中的資料表複本。不過，您不能在將表的複本作為獨立實體保留的同時停止複制，也無法暫停複寫。

Amazon DynamoDB Streams 如何與全域資料表互動？

每個全域資料表都會根據其所有的寫入作業產生獨立串流，無論從何處開始皆然。您可以選擇在一個區域或所有區域中 (獨立) 使用此 DynamoDB 串流。如果您想要處理本機寫入，而不是複寫的寫入操作，可以將自己的區域屬性新增至每個項目。然後，您可以使用 AWS Lambda 事件篩選條件，呼叫 Lambda 函數在本機區域中進行寫入。這有助於插入和更新操作，但不能用於刪除操作。

全域資料表如何處理交易？

交易操作僅在最初寫入發生的區域內提供原子性、一致性、隔離性、耐久性 (ACID) 保證。全域資料表不支援跨區域交易。例如，如果您有一個全域資料表，其在美國東部 (俄亥俄) 和美國西部 (奧勒岡) 區域有複本，並且在美國東部 (俄亥俄) 區域執行 `TransactWriteItems` 操作，在這些變更進行複寫之後，您可能會在美國西部 (奧勒岡) 區域看到部分完成的交易。當變更已在來源區域遞交後，這些變更才會複寫至其他區域。

全域資料表如何與 DynamoDB Accelerator (DAX) 快取互動？

全域資料表會透過直接更新 DynamoDB 來略過 DAX，因此 DAX 不會察覺其持有過時資料。當快取的 TTL 到期時，才會重新整理 DAX 快取。

是否會傳播資料表上的標籤？

否，標籤不會自動傳播。

我應該備份所有區域中的資料表還是只備份一個？

答案是取決於備份的目的。

- 如果您想要確保資料耐久性，則 DynamoDB 已適當提供該保護。該服務確保的就是耐久性。
- 如果您想要保留歷史記錄的快照 (例如，為了符合法規要求)，則在一個區域中進行備份應該就足夠了。您可以使用 AWS Backup 將備份複製到其他區域。
- 如果您想要復原錯誤刪除或修改的資料，請在一個區域中使用 DynamoDB point-in-time 復原 (PITR)。

如何使用 AWS CloudFormation 來部署全域資料表？

- CloudFormation 將 DynamoDB 資料表和全域資料表代表為兩個獨立的資源：`AWS::DynamoDB::Table`和。`AWS::DynamoDB::GlobalTable`其中一種方法是使用 `GlobalTable` 建構模組來建立可能是全域資料表的所有資料表，並在需要時在稍後新增區域。
- 在中 CloudFormation，無論複本數目為何，每個全域表都由單一區域中的單一堆疊控制。部署範本時， CloudFormation 會建立並更新所有複本，做為單一堆疊作業的一部分。您不應該在多個區域中部署相同的 AWS::DynamoDB::GlobalTable 資源。這會導致錯誤且不支援。如果您在多個區域中部署應用程式範本，則可以使用條件在單一區域中建立 `AWS::DynamoDB::GlobalTable` 資源。您也可以選擇在與應用程式分開的堆疊中定義 `AWS::DynamoDB::GlobalTable` 資源，並確保其部署到單一區域。
- 如果您有一個常規表，並且想要將其轉換為全局表，同時保持其管理 CloudFormation：將刪除策略設置為`Retain`，從堆棧中刪除表，將表轉換為控制台中的全局表，然後將全局表作為新資源導入堆棧。如需詳細資訊，請參閱AWS GitHub 儲存庫 amazon-dynamodb-table-to-global-table-cdk。
- 目前不支援跨帳戶複寫。

結論和資源

DynamoDB 全域資料表具有極少的控制項，但仍需要仔細考量。您必須決定寫入模式、路由模型和疏散程序。您必須在每個區域檢測您的應用程式，並準備好調整路由或執行疏散以維護全域狀況。獎勵是具有具有低延遲讀寫操作的全球分佈式數據集，該數據集專為 99.999% 的可用性而設計。

如需 DynamoDB 全域資料表的詳細資訊，請參閱下列資源：

- Amazon DynamoDB 文件
- Amazon 路線 53 應用程序恢復控
- ARC準備檢查（AWS 文檔）
- 路線 53 路由政策（AWS 文件）
- AWS Global Accelerator
- 服務等級協議
- AWS 多區域基礎知識 (AWS 白皮書)
- 資料彈性設計模式與 AWS (AWS 重新:發明 2022 簡報)
- 富達投資與投資如何利用 Amazon DynamoDB 現代化 (RE: 發明 2022 簡報)AWS
- 多區域設計模式和最佳實踐（AWS 重新：發明 2022 演示文稿）
- 開啟災難復原 (DR) 架構 AWS，部分III：指示燈與暖待命 (部AWS 落格文章)
- 使用區域固定功能為 Amazon DynamoDB 全域表中的項目設定主區域 (AWS 部落格文章)
- 監控 Amazon DynamoDB 支援營運感知 (AWS 部落格文章)
- 擴展 DynamoDB：分割區、快速鍵和分割如何影響熱效能 (AWS 部落格文章)

文件歷史紀錄

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
更新 AWS Global Accelerator 資訊	更正「全域加速器」要求路由的端點。	2024年3月14日
更新的 AWS 區域 支援資訊	更新了常見問答集，指出全域資料表現在支援所有 AWS 區域。	2023 年 11 月 15 日
初次出版	—	2023 年 5 月 19 日

AWS 規定指引詞彙

以下是 AWS 規範性指引所提供的策略、指南和模式中常用的術語。若要建議項目，請使用詞彙表末尾的提供意見回饋連結。

數字

7 R

將應用程式移至雲端的七種常見遷移策略。這些策略以 Gartner 在 2011 年確定的 5 R 為基礎，包括以下內容：

- 重構/重新架構 – 充分利用雲端原生功能來移動應用程式並修改其架構，以提高敏捷性、效能和可擴展性。這通常涉及移植作業系統和資料庫。範例：將您的現場部署 Oracle 資料庫遷移到與 Amazon Aurora PostgreSQL 相容的版本。
- 平台轉換 (隨即重塑) – 將應用程式移至雲端，並引入一定程度的優化以利用雲端功能。範例：將您的現場部署 Oracle 資料庫遷移到 Amazon Relational Database Service 服務 (Amazon RDS)，適用於 AWS 雲端.
- 重新購買 (捨棄再購買) – 切換至不同的產品，通常從傳統授權移至 SaaS 模型。範例：將您的客戶關係管理 (CRM) 系統遷移至 SalesForce.com。
- 主機轉換 (隨即轉移) – 將應用程式移至雲端，而不進行任何變更以利用雲端功能。範例：將您的現場部署 Oracle 資料庫遷移至中 EC2 執行個體上的 Oracle 資料庫 AWS 雲端。
- 重新放置 (虛擬機器監視器等級隨即轉移) – 將基礎設施移至雲端，無需購買新硬體、重寫應用程式或修改現有操作。您可以將伺服器從內部部署平台遷移到相同平台的雲端服務。範例：將 Microsoft Hyper-V應用程式移轉至 AWS.
- 保留 (重新檢視) – 將應用程式保留在來源環境中。其中可能包括需要重要重構的應用程式，且您希望將該工作延遲到以後，以及您想要保留的舊版應用程式，因為沒有業務理由來進行遷移。
- 淘汰 – 解除委任或移除來源環境中不再需要的應用程式。

A

ABAC

請參閱以屬性為基礎的存取控制。

抽象的服務

請參閱受管理服務。

酸

請參閱原子性、一致性、隔離性、耐用性。

主動-主動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步 (透過使用雙向複寫工具或雙重寫入操作)，且兩個資料庫都在遷移期間處理來自連接應用程式的交易。此方法支援小型、受控制批次的遷移，而不需要一次性切換。它比主動-被動遷移更具彈性，但需要更多的工作。

主動-被動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步，但只有來源資料庫處理來自連接應用程式的交易，同時將資料複寫至目標資料庫。目標資料庫在遷移期間不接受任何交易。

聚合函數

在一組資料列上運作，並計算群組的單一傳回值的 SQL 函數。彙總函式的範例包括SUM和MAX。

AI

請參閱人工智慧。

艾奧運

請參閱人工智慧作業。

匿名化

永久刪除資料集中個人資訊的程序。匿名化可以幫助保護個人隱私。匿名資料不再被視為個人資料。

反模式

一種經常使用的解決方案，用於解決方案的生產力適得其反，效果不佳或效果低於替代方案。

應用控制

一種安全性方法，只允許使用核准的應用程式，以協助保護系統免受惡意軟體的攻擊。

應用程式組合

有關組織使用的每個應用程式的詳細資訊的集合，包括建置和維護應用程式的成本及其商業價值。此資訊是產品組合探索和分析程序的關鍵，有助於識別要遷移、現代化和優化的應用程式並排定其優先順序。

人工智慧 (AI)

電腦科學領域，致力於使用運算技術來執行通常與人類相關的認知功能，例如學習、解決問題和識別模式。如需詳細資訊，請參閱什麼是人工智慧？

人工智慧操作 (AIOps)

使用機器學習技術解決操作問題、減少操作事件和人工干預以及提高服務品質的程序。如需有關如何在 AWS 遷移策略中使用 AIOps 的詳細資訊，請參閱操作整合指南。

非對稱加密

一種加密演算法，它使用一對金鑰：一個用於加密的公有金鑰和一個用於解密的私有金鑰。您可以共用公有金鑰，因為它不用於解密，但對私有金鑰存取應受到高度限制。

原子性、一致性、隔離性、耐久性 (ACID)

一組軟體屬性，即使在出現錯誤、電源故障或其他問題的情況下，也能確保資料庫的資料有效性和操作可靠性。

屬性型存取控制 (ABAC)

根據使用者屬性 (例如部門、工作職責和團隊名稱) 建立精細許可的實務。如需詳細資訊，請參閱 AWS Identity and Access Management (IAM) 文件 AWS中的 ABAC。

授權資料來源

儲存資料主要版本的位置，被認為是最可靠的資訊來源。您可以將授權資料來源中的資料複製到其他位置，以便處理或修改資料，例如匿名化、編輯或將其虛擬化。

可用區域

一個獨立的位置，與其他 AWS 區域 可用區域中的故障隔離，並為相同區域中的其他可用區域提供廉價、低延遲的網路連線能力。

AWS 雲端採用架構 (AWS CAF)

指導方針和最佳做法的架構，可協 AWS 助組織制定有效率且有效的計畫，以順利移轉至雲端。AWS CAF 將指導組織到六個重點領域，稱為觀點：業務，人員，治理，平台，安全性和運營。業務、人員和控管層面著重於業務技能和程序；平台、安全和操作層面著重於技術技能和程序。例如，人員層面針對處理人力資源 (HR)、人員配備功能和人員管理的利害關係人。針對此觀點，AWS CAF 為人員開發、訓練和通訊提供指導，以協助組織為成功採用雲端做好準備。如需詳細資訊，請參閱 AWS CAF 網站和 AWS CAF 白皮書。

AWS 工作負載資格架構 (AWS WQF)

可評估資料庫移轉工作負載、建議移轉策略並提供工作預估的工具。 AWS WQF 包含在 AWS Schema Conversion Tool ()AWS SCT中。它會分析資料庫結構描述和程式碼物件、應用程式程式碼、相依性和效能特性，並提供評估報告。

B

壞機器人

旨在破壞或對個人或組織造成傷害的機器人。

BCP

請參閱業務連續性規劃。

行為圖

資源行為的統一互動式檢視，以及一段時間後的互動。您可以將行為圖與 Amazon Detective 搭配使用來檢查失敗的登入嘗試、可疑的 API 呼叫和類似動作。如需詳細資訊，請參閱偵測文件中的行為圖中的資料。

大端序系統

首先儲存最高有效位元組的系統。另請參閱 「位元順序」。

二進制分類

預測二進制結果的過程 (兩個可能的類別之一)。例如，ML 模型可能需要預測諸如「此電子郵件是否是垃圾郵件？」等問題 或「產品是書還是汽車？」

Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

藍/綠部署

建立兩個獨立但相同環境的部署策略。您可以在一個環境中執行目前的應用程式版本 (藍色)，而在另一個環境 (綠色) 中執行新的應用程式版本。此策略可協助您以最小的影響快速回復。

機器人

透過網際網路執行自動化工作並模擬人類活動或互動的軟體應用程式。某些漫遊器是有用的或有益的，例如用於索引 Internet 上信息的網絡爬蟲。其他一些機器人（稱為不良機器人）旨在破壞或對個人或組織造成傷害。

殭屍網絡

受惡意軟件感染並受到單一方（稱為機器人牧民或機器人操作員）控制的機器人網絡。殭屍網絡是擴展機器人及其影響的最著名機制。

分支

程式碼儲存庫包含的區域。儲存庫中建立的第一個分支是主要分支。您可以從現有分支建立新分支，然後在新分支中開發功能或修正錯誤。您建立用來建立功能的分支通常稱為功能分支。當準備好發佈功能時，可以將功能分支合併回主要分支。如需詳細資訊，請參閱關於分支 (GitHub 文件)。

防碎玻璃訪問

在特殊情況下，並透過核准的程序，使用者可以快速取得他 AWS 帳戶 們通常沒有存取權限的存取權。如需詳細資訊，請參閱 AWS Well-Architected 指南中的實作防破玻璃程序指標。

棕地策略

環境中的現有基礎設施。對系統架構採用棕地策略時，可以根據目前系統和基礎設施的限制來設計架構。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

緩衝快取

儲存最常存取資料的記憶體區域。

業務能力

業務如何創造價值 (例如，銷售、客戶服務或營銷)。業務能力可驅動微服務架構和開發決策。如需詳細資訊，請參閱在 AWS上執行容器化微服務白皮書的圍繞業務能力進行組織部分。

業務連續性規劃 (BCP)

一種解決破壞性事件 (如大規模遷移) 對營運的潛在影響並使業務能夠快速恢復營運的計畫。

C

咖啡

請參閱AWS 雲端採用架構。

金絲雀部署

向最終用戶發行版本的緩慢和增量版本。當您有信心時，您可以部署新版本並完全取代目前的版本。

CCoE

請參閱雲端卓越中心。

CDC

請參閱變更資料擷取。

變更資料擷取 (CDC)

追蹤對資料來源 (例如資料庫表格) 的變更並記錄有關變更的中繼資料的程序。您可以將 CDC 用於各種用途，例如稽核或複寫目標系統中的變更以保持同步。

混沌工程

故意引入故障或破壞性事件來測試系統的彈性。您可以使用 AWS Fault Injection Service (AWS FIS) 執行實驗來 stress 您的 AWS 工作負載並評估其回應。

CI/CD

請參閱持續整合和持續交付。

分類

有助於產生預測的分類程序。用於分類問題的 ML 模型可預測離散值。離散值永遠彼此不同。例如，模型可能需要評估影像中是否有汽車。

用戶端加密

在目標 AWS 服務 接收資料之前，在本機加密資料。

雲端卓越中心 (CCoE)

一個多學科團隊，可推動整個組織的雲端採用工作，包括開發雲端最佳實務、調動資源、制定遷移時間表以及領導組織進行大規模轉型。如需詳細資訊，請參閱 AWS 雲端 企業策略部落格上的 CCoE 文章。

雲端運算

通常用於遠端資料儲存和 IoT 裝置管理的雲端技術。雲計算通常連接到邊緣計算技術。

雲端運作模式

在 IT 組織中，這是用來建置、成熟和最佳化一或多個雲端環境的作業模型。如需詳細資訊，請參閱建立您的雲端作業模型。

採用雲端階段

組織移轉至下列四個階段時通常會經歷 AWS 雲端：

- 專案 – 執行一些與雲端相關的專案以進行概念驗證和學習用途
- 基礎 – 進行基礎投資以擴展雲端採用 (例如，建立登陸區域、定義 CCoE、建立營運模型)
- 遷移 – 遷移個別應用程式
- 重塑 – 優化產品和服務，並在雲端中創新

這些階段是 Stephen Orban 在 AWS 雲端 企業策略部落格部落格文章「邁向雲端優先的旅程與採用階段」中所定義的。如需其與 AWS 移轉策略之間關聯的詳細資訊，請參閱移轉準備指南。

CMDB

請參閱組態管理資料庫。

程式碼儲存庫

透過版本控制程序來儲存及更新原始程式碼和其他資產 (例如文件、範例和指令碼) 的位置。常見的雲存儲庫包括 GitHub 或 AWS CodeCommit。程式碼的每個版本都稱為分支。在微服務結構中，每個儲存庫都專用於單個功能。單一 CI/CD 管道可以使用多個儲存庫。

冷快取

一種緩衝快取，它是空的、未填充的，或者包含過時或不相關的資料。這會影響效能，因為資料庫執行個體必須從主記憶體或磁碟讀取，這比從緩衝快取讀取更慢。

冷資料

很少存取且通常是歷史資料。查詢此類資料時，通常可以接受緩慢的查詢。將此資料移至效能較低且成本較低的儲存層或類別可降低成本。

計算機視覺（CV）

一個 AI 領域，它使用機器學習來分析和從數字圖像和視頻等視覺格式中提取信息。例如，提 AWS Panorama 供將 CV 添加到現場部署攝像機網絡的設備，Amazon 為 CV SageMaker 提供圖像處理算法。

配置漂移

對於工作負載，組態會從預期的狀態變更。這可能會導致工作負載變得不合規，而且通常是漸進且無意的。

組態管理資料庫 (CMDB)

儲存和管理有關資料庫及其 IT 環境的資訊的儲存庫，同時包括硬體和軟體元件及其組態。您通常在遷移的產品組合探索和分析階段使用 CMDB 中的資料。

一致性套件

AWS Config 規則和補救動作的集合，您可以組合這些動作來自訂合規性和安全性檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶 和區域中的單一實體，或跨組織部署。如需詳細資訊，請參閱文件中的AWS Config 一致性套件。

持續整合和持續交付 (CI/CD)

自動化軟體發行程序的來源、建置、測試、暫存和生產階段的程序。CI/CD 通常被描述為管道。CI/CD 可協助您將程序自動化、提升生產力、改善程式碼品質以及加快交付速度。如需詳細資訊，請參閱持續交付的優點。CD 也可表示持續部署。如需詳細資訊，請參閱持續交付與持續部署。

CV

請參閱電腦視覺。

D

靜態資料

網路中靜止的資料，例如儲存中的資料。

資料分類

根據重要性和敏感性來識別和分類網路資料的程序。它是所有網路安全風險管理策略的關鍵組成部分，因為它可以協助您確定適當的資料保護和保留控制。資料分類是 AWS Well-Architected 架構中安全性支柱的一個組成部分。如需詳細資訊，請參閱資料分類。

資料漂移

生產資料與用來訓練 ML 模型的資料之間有意義的變化，或輸入資料隨著時間的推移有意義的變化。資料漂移可降低機器學習模型預測中的整體品質、準確性和公平性。

傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

資料網格

透過集中式管理和控管，提供分散式、分散式資料擁有權的架構架構。

資料最小化

僅收集和處理絕對必要的數據的原則。在中執行資料最小化 AWS 雲端 可降低隱私權風險、成本和分析碳足跡。

資料周長

您 AWS 環境中的一組預防性護欄，可協助確保只有受信任的身分正在存取來自預期網路的受信任資源。若要取得更多資訊，請參閱〈在上建立資料周長〉 AWS。

資料預先處理

將原始資料轉換成 ML 模型可輕鬆剖析的格式。預處理資料可能意味著移除某些欄或列，並解決遺失、不一致或重複的值。

數據來源

在整個生命週期中追蹤資料來源和歷史記錄的程序，例如資料的產生、傳輸和儲存方式。

資料主體

正在收集和處理資料的個人。

資料倉儲

支援商業智慧 (例如分析) 的資料管理系統。資料倉儲通常包含大量歷史資料，通常用於查詢和分析。

資料庫定義語言 (DDL)

用於建立或修改資料庫中資料表和物件之結構的陳述式或命令。

資料庫處理語言 (DML)

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

DDL

請參閱資料庫定義語言。

深度整體

結合多個深度學習模型進行預測。可以使用深度整體來獲得更準確的預測或估計預測中的不確定性。

深度學習

一個機器學習子領域，它使用多層人工神經網路來識別感興趣的輸入資料與目標變數之間的對應關係。

defense-in-depth

這是一種資訊安全方法，其中一系列的安全機制和控制項會在整個電腦網路中精心分層，以保護網路和其中資料的機密性、完整性和可用性。在上採用此策略時 AWS，您可以在 AWS

Organizations 結構的不同層加入多個控制項，以協助保護資源。例如，一 defense-in-depth 種方法可能會結合多因素驗證、網路分段和加密。

委派的管理員

在中 AWS Organizations，相容的服務可以註冊成 AWS 員帳戶，以管理組織的帳戶並管理該服務的權限。此帳戶稱為該服務的委派管理員。如需詳細資訊和相容服務清單，請參閱 AWS Organizations 文件中的可搭配 AWS Organizations運作的服務。

部署

在目標環境中提供應用程式、新功能或程式碼修正的程序。部署涉及在程式碼庫中實作變更，然後在應用程式環境中建置和執行該程式碼庫。

開發環境

請參閱環境。

偵測性控制

一種安全控制，用於在事件發生後偵測、記錄和提醒。這些控制是第二道防線，提醒您注意繞過現有預防性控制的安全事件。如需詳細資訊，請參閱在 AWS上實作安全控制中的偵測性控制。

發展價值流映射

用於識別限制並排定優先順序，對軟體開發生命週期中的速度和品質產生不利影響的程序。DVSM 擴展了最初為精益生產實踐而設計的價值流映射流程。它著重於創造和通過軟件開發過程中移動價值所需的步驟和團隊。

數字雙胞胎

真實世界系統的虛擬表現法，例如建築物、工廠、工業設備或生產線。數位雙胞胎支援預測性維護、遠端監控和生產最佳化。

維度表

在 star 結構描述中，較小的資料表包含事實資料表中定量資料的相關資料屬性。維度表格屬性通常是文字欄位或離散數字，其行為類似於文字。這些屬性通常用於查詢限制、篩選和結果集標籤。

災難

防止工作負載或系統在其主要部署位置達成其業務目標的事件。這些事件可能是自然災害、技術故障或人為行為造成的結果，例如意外設定錯誤或惡意軟體攻擊。

災難復原 (DR)

您使用的策略和程序，將因災難造成的停機時間和資料遺失降到最低。如需詳細資訊，請參閱 AWS Well-Architected 的架構中的雲端中的工作負載的災難復原 AWS：雲端復原。

DML

請參閱資料庫操作語言。

領域驅動的設計

一種開發複雜軟體系統的方法，它會將其元件與每個元件所服務的不斷發展的領域或核心業務目標相關聯。Eric Evans 在其著作 Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003) 中介紹了這一概念。如需有關如何將領域驅動的設計與 strangler fig 模式搭配使用的資訊，請參閱使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET (ASMX) Web 服務。

博士

請參閱災難復原。

漂移檢測

追蹤基線組態的偏差。例如，您可以用 AWS CloudFormation 來偵測系統資源中的漂移，也可以用 AWS Control Tower 來偵測 landing zone 中可能會影響法規遵循治理要求的變更。

DVSM

請參閱開發價值流映射。

E

EDA

請參閱探索性資料分析。

邊緣運算

提升 IoT 網路邊緣智慧型裝置運算能力的技術。與雲計算相比，邊緣計算可以減少通信延遲並縮短響應時間。

加密

一種計算過程，將純文本數據（這是人類可讀的）轉換為密文。

加密金鑰

由加密演算法產生的隨機位元的加密字串。金鑰長度可能有所不同，每個金鑰的設計都是不可預測且唯一的。

端序

位元組在電腦記憶體中的儲存順序。大端序系統首先儲存最高有效位元組。小端序系統首先儲存最低有效位元組。

端點

請參閱服務端點。

端點服務

您可以在虛擬私有雲端 (VPC) 中託管以與其他使用者共用的服務。您可以使用其他或 (IAM) 主體建立端點服務， AWS PrivateLink 並將權限授予其他 AWS 帳戶 或 AWS Identity and Access Management (IAM) 主體。這些帳戶或主體可以透過建立介面 VPC 端點私下連接至您的端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud (Amazon VPC) 文件中的建立端點服務。

企業資源規劃

可自動化並管理企業關鍵業務流程 (例如會計、MES 和專案管理) 的系統。

信封加密

使用另一個加密金鑰對某個加密金鑰進行加密的程序。如需詳細資訊，請參閱 AWS Key Management Service (AWS KMS) 文件中的信封加密。

環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

- 開發環境 – 執行中應用程式的執行個體，只有負責維護應用程式的核心團隊才能使用。開發環境用來測試變更，然後再將開發環境提升到較高的環境。此類型的環境有時稱為測試環境。
- 較低的環境 – 應用程式的所有開發環境，例如用於初始建置和測試的開發環境。
- 生產環境 – 最終使用者可以存取的執行中應用程式的執行個體。在 CI/CD 管道中，生產環境是最後一個部署環境。
- 較高的環境 – 核心開發團隊以外的使用者可存取的所有環境。這可能包括生產環境、生產前環境以及用於使用者接受度測試的環境。

epic

在敏捷方法中，有助於組織工作並排定工作優先順序的功能類別。epic 提供要求和實作任務的高層級描述。例如， AWS CAF 安全史詩包括身份和訪問管理，偵探控制，基礎結構安全性，數據保護和事件響應。如需有關 AWS 遷移策略中的 Epic 的詳細資訊，請參閱計畫實作指南。

ERP

請參閱企業資源規劃。

探索性資料分析 (EDA)

分析資料集以了解其主要特性的過程。您收集或彙總資料，然後執行初步調查以尋找模式、偵測異常並檢查假設。透過計算摘要統計並建立資料可視化來執行 EDA。

F

事實表

[星型架](#)構中的中央表格。它存儲有關業務運營的定量數據。事實資料表通常包含兩種類型的資料欄：包含計量的資料欄，以及包含維度表格外部索引鍵的資料欄。

快速失敗

一種使用頻繁和增量測試來減少開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離邊界

在中 AWS 雲端，可用區域、 AWS 區域控制平面或資料平面等界限，可限制故障的影響，並協助改善工作負載的彈性。如需詳細資訊，請參閱AWS 錯誤隔離邊界。

功能分支

請參閱分支。

特徵

用來進行預測的輸入資料。例如，在製造環境中，特徵可能是定期從製造生產線擷取的影像。

功能重要性

特徵對於模型的預測有多重要。這通常表示為可以透過各種技術來計算的數值得分，例如 Shapley Additive Explanations (SHAP) 和積分梯度。如需詳細資訊，請參閱機器學習模型可解釋性：AWS。

特徵轉換

優化 ML 程序的資料，包括使用其他來源豐富資料、調整值、或從單一資料欄位擷取多組資訊。這可讓 ML 模型從資料中受益。例如，如果將「2021-05-27 00:15:37」日期劃分為「2021」、「五月」、「週四」和「15」，則可以協助學習演算法學習與不同資料元件相關聯的細微模式。

FGAC

請參閱精細的存取控制。

精細的存取控制 (FGAC)

使用多個條件來允許或拒絕訪問請求。

閃切遷移

一種資料庫移轉方法，透過變更資料擷取使用連續資料複寫，在最短的時間內移轉資料，而不是使用階段化方法。目標是將停機時間降至最低。

G

地理阻塞

請參閱地理限制。

地理限制 (地理封鎖)

在 Amazon 中 CloudFront，防止特定國家/地區的使用者存取內容分發的選項。您可以使用允許清單或封鎖清單來指定核准和禁止的國家/地區。如需詳細資訊，請參閱 CloudFront 文件中的限制內容的地理分佈。

Gitflow 工作流程

這是一種方法，其中較低和較高環境在原始碼儲存庫中使用不同分支。Gitflow 工作流程被認為是遺留的，基於主幹的工作流程是現代的首選方法。

綠地策略

新環境中缺乏現有基礎設施。對系統架構採用綠地策略時，可以選擇所有新技術，而不會限制與現有基礎設施的相容性，也稱為棕地。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

防護機制

有助於跨組織單位 (OU) 來管控資源、政策和合規的高層級規則。預防性防護機制會強制執行政策，以確保符合合規標準。透過使用服務控制政策和 IAM 許可界限來將其實作。偵測性防護機制可偵測政策違規和合規問題，並產生提醒以便修正。它們是通過使用 AWS Config，Amazon AWS Security Hub GuardDuty， AWS Trusted Advisor亞馬遜檢查 Amazon Inspector 和自定義 AWS Lambda 檢查來實現的。

H

公頃

查看高可用性。

異質資料庫遷移

將來源資料庫遷移至使用不同資料庫引擎的目標資料庫 (例如，Oracle 至 Amazon Aurora)。異質遷移通常是重新架構工作的一部分，而轉換結構描述可能是一項複雜任務。AWS 提供有助於結構描述轉換的 AWS SCT。

高可用性 (HA)

工作負載在遇到挑戰或災難時持續運作的能力，無需干預。HA 系統的設計可自動容錯移轉、持續提供高品質的效能，以及處理不同的負載和故障，並將效能影響降到最低。

歷史學家現代化

一種用於現代化和升級操作技術 (OT) 系統的方法，以更好地滿足製造業的需求。歷史學家是一種類型的數據庫，用於收集和存儲工廠中的各種來源的數據。

異質資料庫遷移

將您的來源資料庫遷移至共用相同資料庫引擎的目標資料庫 (例如，Microsoft SQL Server 至 Amazon RDS for SQL Server)。同質遷移通常是主機轉換或平台轉換工作的一部分。您可以使用原生資料庫公用程式來遷移結構描述。

熱數據

經常存取的資料，例如即時資料或最近的轉譯資料。此資料通常需要高效能的儲存層或類別，才能提供快速的查詢回應。

修補程序

緊急修正生產環境中的關鍵問題。由於其緊迫性，修補程式通常是在典型的 DevOps 發行工作流程之外進行。

超級護理期間

在切換後，遷移團隊在雲端管理和監控遷移的應用程式以解決任何問題的時段。通常，此期間的長度為 1-4 天。在超級護理期間結束時，遷移團隊通常會將應用程式的責任轉移給雲端營運團隊。

I

IaC

查看基礎結構即程式碼。

身分型政策

附加至一或多個 IAM 主體的政策，用於定義其在 AWS 雲端 環境中的許可。

閒置應用程式

90 天期間 CPU 和記憶體平均使用率在 5% 至 20% 之間的應用程式。在遷移專案中，通常會淘汰這些應用程式或將其保留在內部部署。

IIoT

請參閱工業物聯網。

不可變基礎設施

為生產工作負載部署新基礎結構的模型，而不是更新、修補或修改現有基礎結構。不可變的基礎架構本質上比可變基礎架構更加一致、可靠且可預測。如需詳細資訊，請參閱 Well-Architected 的架構中的使用不可變基礎結 AWS 構進行部署最佳作法。

傳入 (輸入) VPC

在 AWS 多帳戶架構中，VPC 可接受、檢查和路由來自應用程式外部的網路連線。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

增量遷移

一種切換策略，您可以在其中將應用程式分成小部分遷移，而不是執行單一、完整的切換。例如，您最初可能只將一些微服務或使用者移至新系統。確認所有項目都正常運作之後，您可以逐步移動其他微服務或使用者，直到可以解除委任舊式系統。此策略可降低與大型遷移關聯的風險。

工業 4.0

Klaus Schwab 於 2016 年推出的一個術語，指的是透過連線能力、即時資料、自動化、分析和 AI/ML 的進步來實現製造流程的現代化。

基礎設施

應用程式環境中包含的所有資源和資產。

基礎設施即程式碼 (IaC)

透過一組組態檔案來佈建和管理應用程式基礎設施的程序。IaC 旨在協助您集中管理基礎設施，標準化資源並快速擴展，以便新環境可重複、可靠且一致。

工業物聯網 (IIoT)

在製造業、能源、汽車、醫療保健、生命科學和農業等產業領域使用網際網路連線的感測器和裝置。如需詳細資訊，請參閱建立工業物聯網 (IIoT) 數位轉型策略。

檢查 VPC

在 AWS 多帳戶架構中，集中式 VPC 可管理 VPC（相同或不同 AWS 區域）、網際網路和內部部署網路之間的網路流量檢查。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

物聯網 (IoT)

具有內嵌式感測器或處理器的相連實體物體網路，其透過網際網路或本地通訊網路與其他裝置和系統進行通訊。如需詳細資訊，請參閱什麼是 IoT？

可解釋性

機器學習模型的一個特徵，描述了人類能夠理解模型的預測如何依賴於其輸入的程度。如需詳細資訊，請參閱 AWS 的機器學習模型可解釋性。

IoT

請參閱物聯網。

IT 資訊庫 (ITIL)

一組用於交付 IT 服務並使這些服務與業務需求保持一致的最佳實務。ITIL 為 ITSM 提供了基礎。

IT 服務管理 (ITSM)

與組織的設計、實作、管理和支援 IT 服務關聯的活動。如需有關將雲端操作與 ITSM 工具整合的資訊，請參閱操作整合指南。

ITIL

請參閱 IT 資訊庫。

ITSM

請參閱 IT 服務管理。

L

標籤式存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中每個使用者和資料本身都明確指派一個安全性標籤值。使用者安全性標籤與資料安全性標籤之間的交集決定了使用者可以看到哪些列與欄。

登陸區域

landing zone 是一個架構良好的多帳戶 AWS 環境，具有可擴展性和安全性。這是一個起點，您的組織可以從此起點快速啟動和部署工作負載與應用程式，並對其安全和基礎設施環境充滿信心。如需有關登陸區域的詳細資訊，請參閱設定安全且可擴展的多帳戶 AWS 環境。

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱以標示為基礎的存取控制。

最低權限

授予執行任務所需之最低許可的安全最佳實務。如需詳細資訊，請參閱 IAM 文件中的套用最低權限許可。

隨即轉移

見 7 盧比

小端序系統

首先儲存最低有效位元組的系統。另請參閱 「位元順序」。

較低的環境

請參閱環境。

M

機器學習 (ML)

一種使用演算法和技術進行模式識別和學習的人工智慧。機器學習會進行分析並從記錄的資料 (例如物聯網 (IoT) 資料) 中學習，以根據模式產生統計模型。如需詳細資訊，請參閱機器學習。

主要分支

請參閱分支。

惡意軟體

旨在危及計算機安全性或隱私的軟件。惡意軟件可能會破壞計算機系統，洩漏敏感信息或獲得未經授權的訪問。惡意軟體的例子包括病毒、蠕蟲、勒索軟體、特洛伊木馬程式、間諜軟體和鍵盤記錄程式。

受管理服務

AWS 服務 用於 AWS 操作基礎架構層、作業系統和平台，並且您可以存取端點以儲存和擷取資料。Amazon Simple Storage Service (Amazon S3) 和 Amazon DynamoDB 是受管服務的範例。這些也稱為抽象服務。

製造執行系統

用於跟踪，監控，記錄和控制生產過程的軟件系統，可在現場將原材料轉換為成品。

MAP

請參閱 Migration Acceleration Program。

機制

一個完整的過程，您可以在其中創建工具，推動工具的採用，然後檢查結果以進行調整。機制是一個循環，它加強和改善自己，因為它運行。如需詳細資訊，請參閱 AWS Well-Architected 的架構中建置機制。

成員帳戶

屬於 AWS 帳戶 中組織的管理帳戶以外的所有帳戶 AWS Organizations。一個帳戶一次只能是一個組織的成員。

MES

請參閱製造執行系統。

郵件佇列遙測傳輸 (MQTT)

以發佈/訂閱模式為基礎的輕量型 machine-to-machine (M2M) 通訊協定，適用於資源受限 IoT 裝置。

微服務

一種小型的獨立服務，它可透過定義明確的 API 進行通訊，通常由小型獨立團隊擁有。例如，保險系統可能包含對應至業務能力 (例如銷售或行銷) 或子領域 (例如購買、索賠或分析) 的微服務。微服

務的優點包括靈活性、彈性擴展、輕鬆部署、可重複使用的程式碼和適應力。如需詳細資訊，請參閱使用 AWS 無伺服器服務整合微服務。

微服務架構

一種使用獨立元件來建置應用程式的方法，這些元件會以微服務形式執行每個應用程式程序。這些微服務會使用輕量型 API，透過明確定義的介面進行通訊。此架構中的每個微服務都可以進行更新、部署和擴展，以滿足應用程式特定功能的需求。如需詳細資訊，請參閱上 AWS的實作微服務。

Migration Acceleration Program (MAP)

提供諮詢支援、訓練和服務的 AWS 計畫，協助組織為移轉至雲端建立穩固的營運基礎，並協助抵消移轉的初始成本。MAP 包括用於有條不紊地執行舊式遷移的遷移方法以及一組用於自動化和加速常見遷移案例的工具。

大規模遷移

將大部分應用程式組合依波次移至雲端的程序，在每個波次中，都會以更快的速度移動更多應用程式。此階段使用從早期階段學到的最佳實務和經驗教訓來實作團隊、工具和流程的遷移工廠，以透過自動化和敏捷交付簡化工作負載的遷移。這是 AWS 遷移策略的第三階段。

遷移工廠

可透過自動化、敏捷的方法簡化工作負載遷移的跨職能團隊。移轉工廠團隊通常包括營運、業務分析師和擁有者、移轉工程師、開發人員和 DevOps 專業人員。20% 至 50% 之間的企業應用程式組合包含可透過工廠方法優化的重複模式。如需詳細資訊，請參閱此內容集中的遷移工廠的討論和雲端遷移工廠指南。

遷移中繼資料

有關完成遷移所需的應用程式和伺服器的資訊。每種遷移模式都需要一組不同的遷移中繼資料。移轉中繼資料的範例包括目標子網路、安全性群組和 AWS 帳戶。

遷移模式

可重複的遷移任務，詳細描述遷移策略、遷移目的地以及所使用的遷移應用程式或服務。範例：使 AWS 用應用程式遷移服務將遷移重新託管到 Amazon EC2。

遷移組合評定 (MPA)

這是一種線上工具，可提供驗證要移轉至的商業案例的 AWS 雲端資訊。MPA 提供詳細的組合評定 (伺服器適當規模、定價、總體擁有成本比較、遷移成本分析) 以及遷移規劃 (應用程式資料分析和資料收集、應用程式分組、遷移優先順序，以及波次規劃)。所有 AWS 顧問和 APN 合作夥伴顧問均可免費使用 MPA 工具 (需要登入)。

遷移準備程度評定 (MRA)

使用 AWS CAF 獲得有關組織雲端準備狀態的見解、識別優勢和弱點，以及建立行動計劃以縮小已識別差距的過程。如需詳細資訊，請參閱遷移準備程度指南。MRA 是 AWS 遷移策略的第一階段。

遷移策略

將工作負載移轉至 AWS 雲端. 如需詳細資訊，請參閱本詞彙表中的 7 Rs 項目，並參閱動員您的組織以加速大規模移轉。

機器學習 (ML)

請參閱機器學習。

現代化

將過時的 (舊版或單一) 應用程式及其基礎架構轉換為雲端中靈活、富有彈性且高度可用的系統，以降低成本、提高效率並充分利用創新。如需詳細資訊，請參閱. AWS 雲端

現代化準備程度評定

這項評估可協助判斷組織應用程式的現代化準備程度；識別優點、風險和相依性；並確定組織能夠在多大程度上支援這些應用程式的未來狀態。評定的結果就是目標架構的藍圖、詳細說明現代化程序的開發階段和里程碑的路線圖、以及解決已發現的差距之行動計畫。如需詳細資訊，請參閱評估應用程式的現代化準備程度 AWS 雲端。

單一應用程式 (單一)

透過緊密結合的程序作為單一服務執行的應用程式。單一應用程式有幾個缺點。如果一個應用程式功能遇到需求激增，則必須擴展整個架構。當程式碼庫增長時，新增或改進單一應用程式的功能也會變得更加複雜。若要解決這些問題，可以使用微服務架構。如需詳細資訊，請參閱將單一體系分解為微服務。

MPA

請參閱移轉組合評估。

MQTT

請參閱佇列遙測傳輸的郵件。

多類別分類

一個有助於產生多類別預測的過程 (預測兩個以上的結果之一)。例如，機器學習模型可能會詢問「此產品是書籍、汽車還是電話？」 或者「這個客戶對哪種產品類別最感興趣？」

可變的基礎

一種模型，用於更新和修改生產工作負載的現有基礎結構。為了提高一致性，可靠性和可預測性，AWS Well-Architected 框架建議使用不可變的基礎結構作為最佳實踐。

O

OAC

請參閱原始存取控制。

OAI

請參閱原始存取身分。

OCM

請參閱組織變更管理。

離線遷移

一種遷移方法，可在遷移過程中刪除來源工作負載。此方法涉及延長停機時間，通常用於小型非關鍵工作負載。

OI

請參閱作業整合。

OLA

請參閱作業層級協定。

線上遷移

一種遷移方法，無需離線即可將來源工作負載複製到目標系統。連接至工作負載的應用程式可在遷移期間繼續運作。此方法涉及零至最短停機時間，通常用於關鍵的生產工作負載。

OPCA

請參閱開放程序通訊-統一架構。

開放程序通訊-統一架構 (OPC-UA)

用於工業自動化的 machine-to-machine (M2M) 通訊協定。OPC-UA 提供數據加密，身份驗證和授權方案的互操作性標準。

操作水準協議 (OLA)

一份協議，闡明 IT 職能群組承諾向彼此提供的內容，以支援服務水準協議 (SLA)。

操作準備程度檢討 (ORR)

問題和相關最佳做法的檢查清單，可協助您瞭解、評估、預防或減少事件和可能的故障範圍。如需詳細資訊，請參閱 AWS Well-Architected 的架構中的作業準備檢閱 (ORR)。

操作技術

可與實體環境搭配使用的硬體和軟體系統，以控制工業作業、設備和基礎設施。在製造業中，整合 OT 和資訊技術 (IT) 系統是工業 4.0 轉型的關鍵焦點。

操作整合 (OI)

在雲端中將操作現代化的程序，其中包括準備程度規劃、自動化和整合。如需詳細資訊，請參閱操作整合指南。

組織追蹤

由建立的追蹤 AWS CloudTrail 記錄中組織 AWS 帳戶 中所有人的所有事件 AWS Organizations。在屬於組織的每個 AWS 帳戶 中建立此追蹤，它會跟蹤每個帳戶中的活動。如需詳細資訊，請參閱CloudTrail文件中的為組織建立追蹤。

組織變更管理 (OCM)

用於從人員、文化和領導力層面管理重大、顛覆性業務轉型的架構。OCM 透過加速變更採用、解決過渡問題，以及推動文化和組織變更，協助組織為新系統和策略做好準備，並轉移至新系統和策略。在 AWS 移轉策略中，這個架構稱為人員加速，因為雲端採用專案所需的變更速度。如需詳細資訊，請參閱 OCM 指南。

原始存取控制 (OAC)

在中 CloudFront，限制存取權限以保護 Amazon Simple Storage Service (Amazon S3) 內容的增強選項。OAC 支援所有 S3 儲存貯體 AWS 區域、伺服器端加密 AWS KMS (SSE-KMS)，以及 S3 儲存貯體的動態PUT和DELETE請求。

原始存取身分 (OAI)

在中 CloudFront，用於限制存取以保護 Amazon S3 內容的選項。當您使用 OAI 時， CloudFront 會建立 Amazon S3 可用來進行驗證的主體。經驗證的主體只能透過特定散發存取 S3 儲存 CloudFront 貯體中的內容。另請參閱 OAC，它可提供更精細且增強的存取控制。

ORR

請參閱作業整備檢閱。

OT

請參閱操作技術。

傳出 (輸出) VPC

在 AWS 多帳戶架構中，處理從應用程式內啟動的網路連線的 VPC。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

P

許可界限

附接至 IAM 主體的 IAM 管理政策，可設定使用者或角色擁有的最大許可。如需詳細資訊，請參閱 IAM 文件中的許可界限。

個人識別資訊 (PII)

直接查看或與其他相關數據配對時，可用於合理推斷個人身份的信息。PII 的範例包括姓名、地址和聯絡資訊。

PII

請參閱個人識別資訊。

手冊

一組預先定義的步驟，可擷取與遷移關聯的工作，例如在雲端中提供核心操作功能。手冊可以採用指令碼、自動化執行手冊或操作現代化環境所需的程序或步驟摘要的形式。

公司

請參閱可編程邏輯控制器

PLM

查看產品生命週期管理。

政策

可以定義權限 (請參閱以身分識別為基礎的策略)、指定存取條件 (請參閱以資源為基礎的策略) 或定義組織中所有帳戶的最大權限的物件 AWS Organizations (請參閱服務控制策略)。

混合持久性

根據資料存取模式和其他需求，獨立選擇微服務的資料儲存技術。如果您的微服務具有相同的資料儲存技術，則其可能會遇到實作挑戰或效能不佳。如果微服務使用最適合其需求的資料儲存，則可以更輕鬆地實作並達到更好的效能和可擴展性。如需詳細資訊，請參閱在微服務中啟用資料持久性。

組合評定

探索、分析應用程式組合並排定其優先順序以規劃遷移的程序。如需詳細資訊，請參閱評估遷移準備程度。

述詞

傳回`true`或的查詢條件`false`，通常位於子`WHERE`句中。

謂詞下推

一種資料庫查詢最佳化技術，可在傳輸前篩選查詢中的資料。這樣可減少必須從關聯式資料庫擷取和處理的資料量，並改善查詢效能。

預防性控制

旨在防止事件發生的安全控制。這些控制是第一道防線，可協助防止對網路的未經授權存取或不必要變更。如需詳細資訊，請參閱在 AWS上實作安全控制中的預防性控制。

委託人

中 AWS 可執行動作和存取資源的實體。此實體通常是 IAM 角色或使用者的根使用者。 AWS 帳戶如需詳細資訊，請參閱 IAM 文件中角色術語和概念中的主體。

隱私設計

一種系統工程方法，在整個工程過程中將隱私權納入考量。

私有託管區域

一種容器，它包含有關您希望 Amazon Route 53 如何回應一個或多個 VPC 內的域及其子域之 DNS 查詢的資訊。如需詳細資訊，請參閱 Route 53 文件中的使用私有託管區域。

主動控制

一種安全控制項，旨在防止部署不符合規範的資源。這些控制項會在資源佈建之前進行掃描。如果資源不符合控制項，則不會佈建該資源。如需詳細資訊，請參閱 AWS Control Tower 文件中的控制項參考指南，並參閱實作安全性控制中的主動控制 AWS。

產品生命週期管理 (PLM)

在產品的整個生命週期中管理資料和流程，從設計、開發、上市到成長與成熟度，再到下降和移除。

生產環境

請參閱環境。

可編程邏輯控制器 (PLC)

在製造業中，一台高度可靠且適應性強的計算機，可監控機器並自動化製造過程。

化名化

以預留位置值取代資料集中的個人識別碼的程序。化名化有助於保護個人隱私。假名化數據仍被認為是個人數據。

發布/訂閱 (發布/訂閱)

一種模式，可在微服務之間實現非同步通訊，以提高延展性和回應能力 例如，在微服務型 MES 中，微服務可以將事件訊息發佈到其他微服務可訂閱的通道。系統可以在不變更發佈服務的情況下新增微服務。

Q

查詢計劃

一系列步驟，如指示，用來存取 SQL 關聯式資料庫系統中的資料。

查詢計劃迴歸

在資料庫服務優化工具選擇的計畫比對資料庫環境進行指定的變更之前的計畫不太理想時。這可能因為對統計資料、限制條件、環境設定、查詢參數繫結的變更以及資料庫引擎的更新所導致。

R

拉齐矩阵

請參閱負責任，負責，諮詢，通知（RACI）。

勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

拉西矩陣

請參閱負責任，負責，諮詢，通知（RACI）。

RCAC

請參閱列與欄存取控制。

僅供讀取複本

用於唯讀用途的資料庫複本。您可以將查詢路由至僅供讀取複本以減少主資料庫的負載。

重新建築師

見 7 盧比

復原點目標 (RPO)

自上次資料復原點以來可接受的時間上限。這決定了最後一個恢復點和服務中斷之間可接受的數據丟失。

復原時間目標 (RTO)

服務中斷與恢復服務之間的最大可接受延遲。

重構

見 7 盧比

區域

地理區域中的 AWS 資源集合。每個都 AWS 區域 是隔離和獨立於其他的，以提供容錯能力，穩定性和彈性。如需詳細資訊，請參閱指定 AWS 區域 您的帳戶可以使用的項目。

迴歸

預測數值的 ML 技術。例如，為了解決「這房子會賣什麼價格？」的問題 ML 模型可以使用線性迴歸模型，根據已知的房屋事實 (例如，平方英尺) 來預測房屋的銷售價格。

重新主持

見 7 盧比

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新定位

見 7 盧比

再平台

見 7 盧比

買回

見 7 盧比

彈性

應用程式抵抗或從中斷中復原的能力。在規劃備援時，高可用性和災難復原是常見的考量因素。AWS 雲端如需詳細資訊，請參閱AWS 雲端 復原力。

資源型政策

附接至資源的政策，例如 Amazon S3 儲存貯體、端點或加密金鑰。這種類型的政策會指定允許存取哪些主體、支援的動作以及必須滿足的任何其他條件。

負責者、當責者、事先諮詢者和事後告知者 (RACI) 矩陣

定義移轉活動和雲端作業所涉及之所有各方的角色與責任的矩陣。矩陣名稱衍生自矩陣中定義的責任型別：負責 (R)、負責 (A)、諮詢 (C) 及通知 (I)。支撐（S）類型是可選的。如果您包含支援，則該矩陣稱為 RASCI 矩陣，如果您將其排除，則稱為 R ACI 矩陣。

回應性控制

一種安全控制，旨在驅動不良事件或偏離安全基準的補救措施。如需詳細資訊，請參閱在 AWS上實作安全控制中的回應性控制。

保留

見 7 盧比

退休

見 7 盧比

旋轉

定期更新密碼以使攻擊者更難以存取認證的程序。

資料列與資料行存取控制 (RCAC)

使用已定義存取規則的基本、彈性 SQL 運算式。RCAC 由資料列權限和資料行遮罩所組成。

RPO

請參閱復原點目標。

RTO

請參閱復原時間目標。

執行手冊

執行特定任務所需的一組手動或自動程序。這些通常是為了簡化重複性操作或錯誤率較高的程序而建置。

S

SAML 2.0

許多身份提供者（IdPs）使用的開放標準。此功能可啟用聯合單一登入 (SSO)，因此使用者可以登入 AWS Management Console 或呼叫 AWS API 作業，而不必為組織中的每個人在 IAM 中建立使用者。如需有關以 SAML 2.0 為基礎的聯合詳細資訊，請參閱 IAM 文件中的關於以 SAML 2.0 為基礎的聯合。

斯卡達

請參閱監督控制和資料擷取。

SCP

請參閱服務控制策略。

秘密

您以加密形式儲存的機密或受限制資訊，例如密碼或使用者認證。 AWS Secrets Manager它由秘密值及其中繼資料組成。密碼值可以是二進位、單一字串或多個字串。如需詳細資訊，請參閱「Secrets Manager 碼中有什麼內容？ 在 Secrets Manager 文檔中。

安全控制

一種技術或管理防護機制，它可預防、偵測或降低威脅行為者利用安全漏洞的能力。安全性控制有四種主要類型：預防性、偵測、回應式和主動式。

安全強化

減少受攻擊面以使其更能抵抗攻擊的過程。這可能包括一些動作，例如移除不再需要的資源、實作授予最低權限的安全最佳實務、或停用組態檔案中不必要的功能。

安全資訊與事件管理 (SIEM) 系統

結合安全資訊管理 (SIM) 和安全事件管理 (SEM) 系統的工具與服務。SIEM 系統會收集、監控和分析來自伺服器、網路、裝置和其他來源的資料，以偵測威脅和安全漏洞，並產生提醒。

安全回應自動化

預先定義且程式化的動作，其設計用來自動回應或修復安全性事件。這些自動化作業可做為偵探或回應式安全控制項，協助您實作 AWS 安全性最佳實務。自動回應動作的範例包括修改 VPC 安全群組、修補 Amazon EC2 執行個體或輪換登入資料。

伺服器端加密

在其目的地的數據加密，通 AWS 服務 過接收它。

服務控制政策 (SCP)

為 AWS Organizations中的組織的所有帳戶提供集中控制許可的政策。SCP 會定義防護機制或設定管理員可委派給使用者或角色的動作限制。您可以使用 SCP 作為允許清單或拒絕清單，以指定允許或禁止哪些服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中的服務控制原則。

服務端點

的進入點的 URL AWS 服務。您可以使用端點，透過程式設計方式連接至目標服務。如需詳細資訊，請參閱 AWS 一般參考 中的 AWS 服務 端點。

服務水準協議 (SLA)

一份協議，闡明 IT 團隊承諾向客戶提供的服務，例如服務正常執行時間和效能。

服務等級指示器 (SLI)

對服務效能層面的測量，例如錯誤率、可用性或輸送量。

服務等級目標 (SLO)

代表服務狀況的目標測量結果，由服務層次指示器測量。

共同責任模式

描述您在雲端安全性和合規方面共享的責任的模型。 AWS AWS 負責雲端的安全性，而您則負責雲端的安全性。如需詳細資訊，請參閱共同責任模式。

暹

請參閱安全性資訊和事件管理系統。

單點故障 (SPF)

應用程式的單一重要元件發生故障，可能會中斷系統。

SLA

請參閱服務等級協議。

SLI

請參閱服務層級指示器。

SLO

請參閱服務等級目標。

split-and-seed 模型

擴展和加速現代化專案的模式。定義新功能和產品版本時，核心團隊會進行拆分以建立新的產品團隊。這有助於擴展組織的能力和服務，提高開發人員生產力，並支援快速創新。如需詳細資訊，請參閱中的應用程式現代化的階段化方法。 AWS 雲端

痙攣

請參閱單一故障點。

星型綱要

使用一個大型事實資料表來儲存交易或測量資料，並使用一或多個較小的維度表格來儲存資料屬性的資料庫組織結構。這種結構是專為在數據倉庫中使用或用於商業智能目的。

Strangler Fig 模式

一種現代化單一系統的方法，它會逐步重寫和取代系統功能，直到舊式系統停止使用為止。此模式源自無花果藤，它長成一棵馴化樹並最終戰勝且取代了其宿主。該模式由 Martin Fowler 引入，作為重寫單一系統時管理風險的方式。如需有關如何套用此模式的範例，請參閱使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET (ASMX) Web 服務。

子網

您 VPC 中的 IP 地址範圍。子網必須位於單一可用區域。

監督控制與資料擷取 (SCADA)

在製造業中，使用硬體與軟體來監控實體資產與生產作業的系統。

對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

合成測試

以模擬使用者互動以偵測潛在問題或監控效能的方式測試系統。您可以使用 Amazon CloudWatch Synthetics 來創建這些測試。

T

標籤

作為組織 AWS 資源的中繼資料的索引鍵值配對。標籤可協助您管理、識別、組織、搜尋及篩選資源。如需詳細資訊，請參閱標記您的 AWS 資源。

目標變數

您嘗試在受監督的 ML 中預測的值。這也被稱為結果變數。例如，在製造設定中，目標變數可能是產品瑕疵。

任務清單

用於透過執行手冊追蹤進度的工具。任務清單包含執行手冊的概觀以及要完成的一般任務清單。對於每個一般任務，它包括所需的預估時間量、擁有者和進度。

測試環境

請參閱環境。

訓練

為 ML 模型提供資料以供學習。訓練資料必須包含正確答案。學習演算法會在訓練資料中尋找將輸入資料屬性映射至目標的模式 (您想要預測的答案)。它會輸出擷取這些模式的 ML 模型。可以使用 ML 模型，來預測您不知道的目標新資料。

傳輸閘道

可以用於互連 VPC 和內部部署網路的網路傳輸中樞。如需詳細資訊，請參閱 AWS Transit Gateway 文件中的傳輸閘道是什麼。

主幹型工作流程

這是一種方法，開發人員可在功能分支中本地建置和測試功能，然後將這些變更合併到主要分支中。然後，主要分支會依序建置到開發環境、生產前環境和生產環境中。

受信任的存取權

授與權限給您指定的服務，以代表您在組織內 AWS Organizations 及其帳戶中執行工作。受信任的服務會在需要該角色時，在每個帳戶中建立服務連結角色，以便為您執行管理工作。如需詳細資訊，請參閱 AWS Organizations 文件中的AWS Organizations 與其他 AWS 服務搭配使用。

調校

變更訓練程序的各個層面，以提高 ML 模型的準確性。例如，可以透過產生標籤集、新增標籤、然後在不同的設定下多次重複這些步驟來訓練 ML 模型，以優化模型。

雙比薩團隊

一個小 DevOps 團隊，你可以餵兩個比薩餅。雙披薩團隊規模可確保軟體開發中的最佳協作。

U

不確定性

這是一個概念，指的是不精確、不完整或未知的資訊，其可能會破壞預測性 ML 模型的可靠性。有兩種類型的不確定性：認知不確定性是由有限的、不完整的資料引起的，而隨機不確定性是由資料中固有的噪聲和隨機性引起的。如需詳細資訊，請參閱量化深度學習系統的不確定性指南。

無差別的任務

也稱為繁重工作，是創建和操作應用程序所必需的工作，但不能為最終用戶提供直接價值或提供競爭優勢。無差異化作業的範例包括採購、維護和容量規劃。

較高的環境

請參閱環境。

V

清空

一種資料庫維護操作，涉及增量更新後的清理工作，以回收儲存並提升效能。

版本控制

追蹤變更的程序和工具，例如儲存庫中原始程式碼的變更。

VPC 對等互連

兩個 VPC 之間的連線，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱 Amazon VPC 文件中的什麼是 VPC 對等互連。

漏洞

會危及系統安全性的軟體或硬體瑕疵。

W

暖快取

包含經常存取的目前相關資料的緩衝快取。資料庫執行個體可以從緩衝快取讀取，這比從主記憶體或磁碟讀取更快。

溫暖的數據

不常存取的資料。查詢此類資料時，通常可以接受中度緩慢的查詢。

視窗功能

一種 SQL 函數，可對以某種方式與當前記錄相關的一組行執行計算。視窗函數對於處理工作非常有用，例如計算移動平均值或根據目前列的相對位置存取列的值。

工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

工作串流

遷移專案中負責一組特定任務的功能群組。每個工作串流都是獨立的，但支援專案中的其他工作串流。例如，組合工作串流負責排定應用程式、波次規劃和收集遷移中繼資料的優先順序。組合工作串流將這些資產交付至遷移工作串流，然後再遷移伺服器和應用程式。

蠕蟲

看到寫一次，多讀。

WQF

請參閱 AWS 工作負載鑑定架構。

寫一次，多讀（WORM）

一種儲存模型，可單次寫入資料並防止資料遭到刪除或修改。授權用戶可以根據需要多次讀取數據，但無法更改數據。這種數據存儲基礎設施被認為是不可變的。

Z

零日漏洞

一種利用零時差漏洞的攻擊，通常是惡意軟件。

零時差漏洞

生產系統中未緩解的瑕疵或弱點。威脅參與者可以利用這種類型的漏洞攻擊系統。由於攻擊，開發人員經常意識到該漏洞。

殭屍應用程式

CPU 和記憶體平均使用率低於 5% 的應用程式。在遷移專案中，通常會淘汰這些應用程式。

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。