使用 Amazon DynamoDB 全域資料表

AWS 方案指引

AWS 方案指引: 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 3 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選擇的 DynamoDB 資料表 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 AWS 多區域部署的更廣泛內容，如AWS 多區域基礎白皮書和具有影片的資料彈性設計模式 AWS所述。

內容

全域資料表概觀

關鍵事實

- 全球資料表有兩種版本：2017.11.29 版（舊版）（有時稱為 v1) 和 2019.11.21 版（目前）（有時稱為 v2)。本指南僅著重於目前的版本。
- 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，且對基礎設施故障具有內部彈性，包括整個可用區域的故障。單一區域 DynamoDB 資料表專為 99.99% 可用性而設計。如需詳細資訊，請參閱 DynamoDB 服務層級協議 (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 資料表專為 99.999% 可用性而設計。透過適當的規劃，全域資料表可協助建立可應對區域性故障的架構。
- 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。收到寫入請求後，本機複本資料表會將寫入操作複寫到背景中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無法一起複寫。
- 來源區域中的每個資料表分割區會平行複寫其寫入操作，並與其他分割區平行。遠端區域中的寫入操作序列可能與來源區域中發生的寫入操作序列不相符。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。其計算方式是查看抵達項目、比較其抵達時間和初始寫入時間，以及計算平均值。計時會儲存在來源區域的 CloudWatch 當中。檢視平均和最大計時對於判斷平均和最壞情況的複寫延遲很有用。此延遲沒有 SLA。
- 如果個別項目在兩個不同區域中大約同時（在此`ReplicationLatency`時段內）更新，且第二個寫入操作發生在第一個寫入操作複寫之前，則可能存在寫入衝突。全域資料表會根據寫入操作的時間戳記，使用最後一個寫入器來獲得機制，以解決此類衝突。第一個操作「遺失」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。最後一個寫入器獲勝方法的實作方式是使用條件式寫入操作，要求傳入項目的時間戳記大於現有項目的時間戳記。
- 全域資料表會將所有項目複寫到所有參與區域。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，並為每個資料表指派不同的參與區域。
- 即使複本區域離線或`ReplicationLatency`成長，本機區域也接受寫入操作。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，所有待處理的傳出和傳入複寫都會重試。不需要特殊動作即可使資料表恢復同步。最後一個寫入器會贏得機制，確保資料最終一致。
- 您可以隨時將新區域新增至 DynamoDB 資料表。DynamoDB 會處理初始同步和持續複寫。您也可以移除區域（甚至是原始區域），這會刪除該區域中的本機資料表。
- DynamoDB 沒有全域端點。所有請求都會對區域端點提出，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。最佳實務是讓託管到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 (DynamoDB 層或周圍堆疊） 內偵測到問題，應將最終使用者流量路由到在不同區域中託管的不同應用程式端點。全域資料表可確保位於每個區域中的應用程式可存取相同的資料。

使用案例

全域資料表提供以下常見優點：

- 低延遲讀取操作。您可以將資料副本放在更接近最終使用者的位置，以減少讀取操作期間的網路延遲。資料會保持與`ReplicationLatency`值一樣新鮮。
- 低延遲寫入操作。最終使用者可以寫入附近的區域，以減少網路延遲和完成寫入操作的時間。必須謹慎路由寫入流量，以確保沒有衝突。下一節將討論路由的技術。
- 提高彈性和災難復原。如果區域效能降低或完全中斷，您可以將其疏散（移出部分或所有前往該區域的請求），並滿足以秒為單位的復原點目標 (RPO) 和復原時間目標 (RTO)。使用全域資料表也會將 DynamoDB SLA 的每月運作時間百分比從 99.99% 增加到 99.999%。
- 無縫區域遷移。您可以新增區域，然後刪除舊區域，將部署從一個區域遷移到另一個區域，而不會在資料層停機。

例如，Re：Invent 2022 中介紹的 Fidelity 投資，其如何使用 DynamoDB 全域資料表作為其訂單管理系統。他們的目標是在無法達到內部部署處理同時維持可用區域和區域故障彈性的規模上，實現可靠低延遲處理。

全域資料表的寫入模式

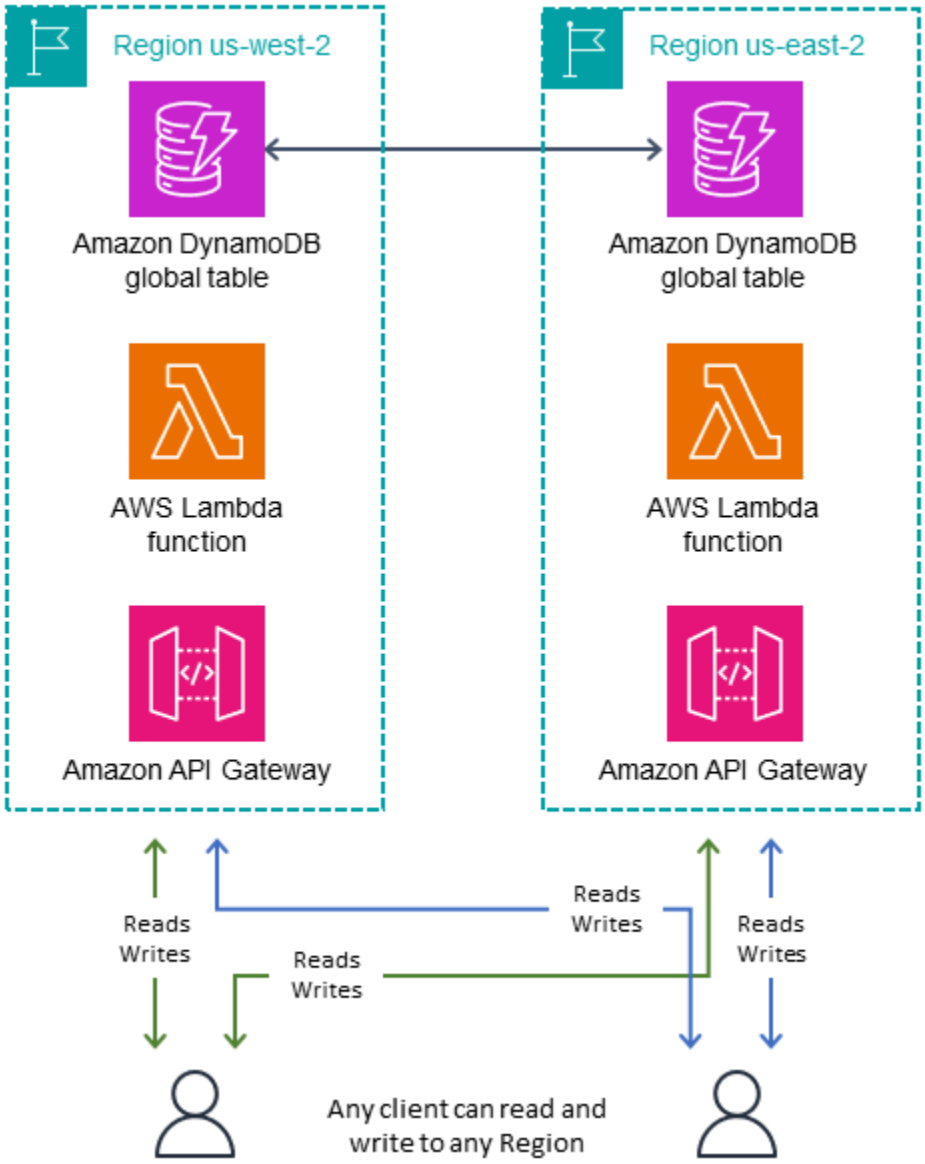
全域表在資料表永遠處於主動-主動資料表層級。不過，您可能想要透過控制路由，將它們用作主動-被動的方式。例如，您可能決定將寫入請求路由到單一區域，以避免潛在的寫入衝突。

有三種主要受管寫入模式，如以下三個章節所述。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。稍後各節中的指引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都是完全作用中的，不會限制寫入操作可能發生的位置。任何區域都可以隨時接受寫入請求。這是最簡單的模式；不過，它只能與某些類型的應用程式搭配使用。當所有寫入操作都具有等冪時，就適合使用。閒置表示它們可安全重複，因此跨區域並行或重複寫入操作不會發生衝突，例如，當使用者更新其聯絡資料時。它也適用於僅附加的資料集，其中所有寫入操作都是確定性主索引鍵下的唯一插入，這是具有等冪能力的特殊情況。最後，此模式適用於可接受寫入操作衝突的風險。



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。容錯移轉更容易，因為任何最近的寫入操作都可以在任何次要區域重播任意次數。盡可能之下，您應該以此為寫入模式進行設計。

例如，數個影片串流服務使用全域資料表來追蹤書籤、檢閱、監看狀態旗標等。這些部署可以使用寫入至任何區域模式，只要它們確保每個寫入操作都是等冪的。如果每次更新 - 例如，設定新的最新時間碼、指派新的檢閱，或設定新的監看狀態 - 直接指派使用者的新狀態，且項目的下一個正確值不取決於其目前值，就會發生這種情況。如果使用者的寫入請求被路由到不同的區域，則上次寫入操作將保留，而全域狀態將根據上次指派進行和解。此模式中的讀取操作最終會保持一致，並會因最新`ReplicationLatency`值而延遲。

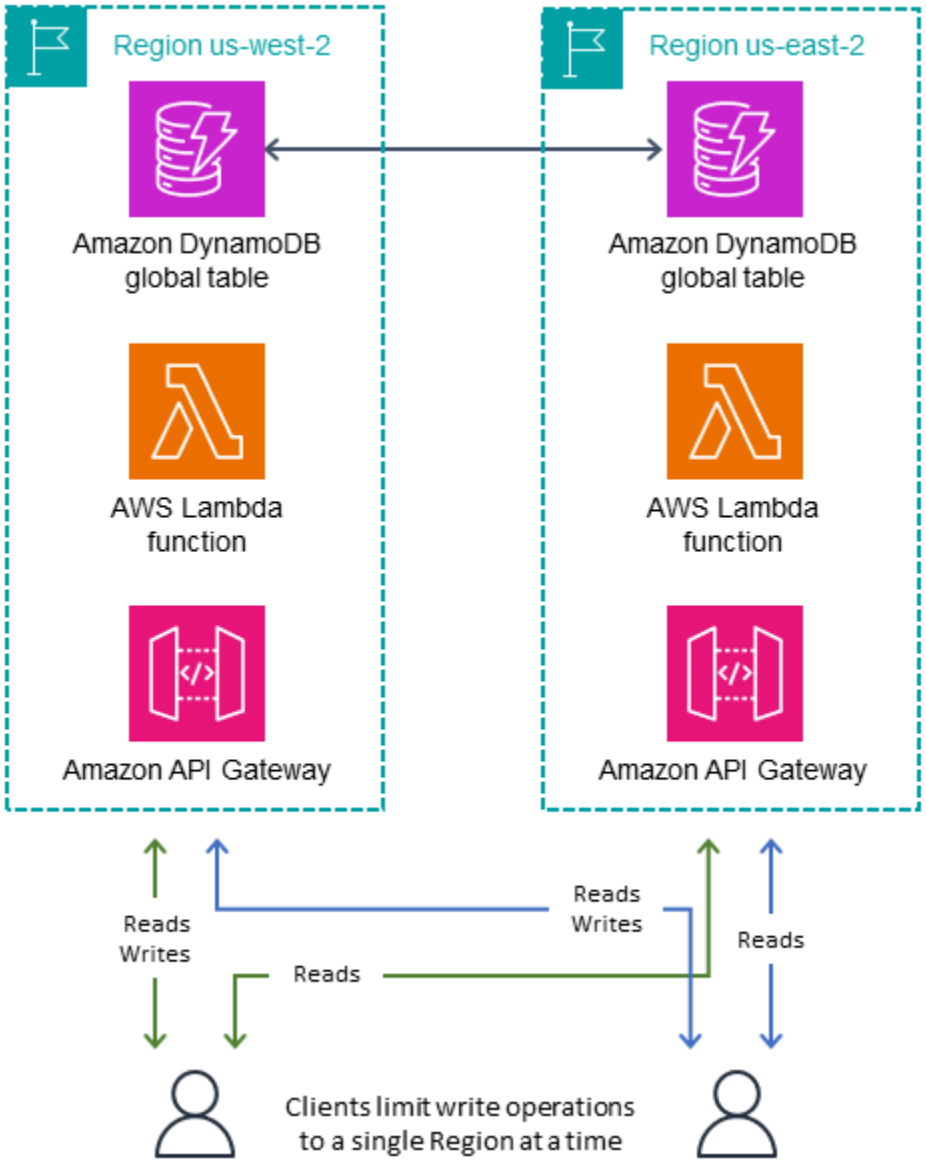
在另一個範例中，一家金融服務公司使用全域資料表作為系統的一部分來維護每個客戶使用借記卡購買的動作記錄，以計算該客戶的現金回饋獎勵。新的交易會從世界各地進行串流並前往多個區域。此公司能夠使用寫入至任何區域模式，並仔細重新設計。初始設計草圖會維護每個客戶的單一RunningBalance項目。客戶動作會使用非等冪的ADD表達式更新平衡（因為新的正確值取決於目前的值），而且如果在不同區域中有兩次寫入操作同時達到相同平衡，則平衡會不同步。重新設計使用事件串流，其運作方式類似具有僅附加工作流程的分類帳。每個客戶動作都會將新項目新增到為該客戶維護的項目收集器。(項目集合是共用主索引鍵但具有不同排序索引鍵的一組項目。) 每個寫入操作都是一個等冪插入，使用客戶 ID 做為分割區索引鍵，而交易 ID 做為排序索引鍵。此設計會讓平衡計算更困難，因為它需要 Query提取項目，後面接著一些用戶端數學，但它讓所有寫入操作都具有同冪性，並在路由和容錯移轉中實現顯著簡化。（本指南稍後會更詳細討論此部分。）

第三個範例涉及提供線上廣告放置服務的公司。此公司決定低資料遺失風險是可以接受的，以實現寫入任何區域模式的設計簡化。當他們提供廣告時，他們只有幾毫秒的時間來擷取足夠的中繼資料，以決定要顯示哪個廣告，然後記錄廣告印模，讓他們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最終使用者的低延遲讀取操作和低延遲寫入操作。它們會在單一項目中記錄使用者的所有廣告印模，以不斷增長的清單表示。他們使用一個項目，而不是附加到項目集合，因此他們可以在每次寫入操作中移除舊廣告印模，而無需支付刪除操作的費用。此寫入操作不具有等冪性；如果同一個最終使用者在大約相同時間看到多個區域中的廣告，則廣告印模的一個寫入操作可能會覆寫另一個。風險是使用者可能會看到廣告一陣子重複一次。他們決定這是可以接受的。

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是主動被動的，並將所有資料表寫入操作路由至單一作用中區域。(DynamoDB 沒有單一作用中區域的概念；DynamoDB 外部的 layer 會管理此概念。) 寫入至一個區域模式可確保寫入操作一次只能流至一個區域，以避免寫入衝突。當您想要使用條件式表達式或交易時，此寫入模式會有所幫助。除非您知道您正在對最新資料採取行動，否則這些表達式是不可能的，因此它們需要將所有寫入請求傳送到具有最新資料的單一區域。

最後，一致的讀取操作可以前往任何複本區域，以達到較低的延遲。強烈一致的讀取操作必須移至單一主要區域。



有時需要變更作用中區域以回應區域故障，如稍後所討論。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。這會將作用中區域放在活動最多的地理位置（通常是白天，因此是名稱）附近，這會導致最低的延遲讀取和寫入操作。它還具有每天呼叫區域變更程式碼的附帶優勢，並確保它在任何災難復原之前都經過良好的測試。

被動區域可能會保留 DynamoDB 周圍的縮減規模基礎設施，只有在它成為作用中區域時才會建立。本指南不包含指示燈和暖待命設計。如需詳細資訊，請參閱部落格文章的災難復原 (DR) 架構 AWS，第 III 部分：指示燈和暖待命。

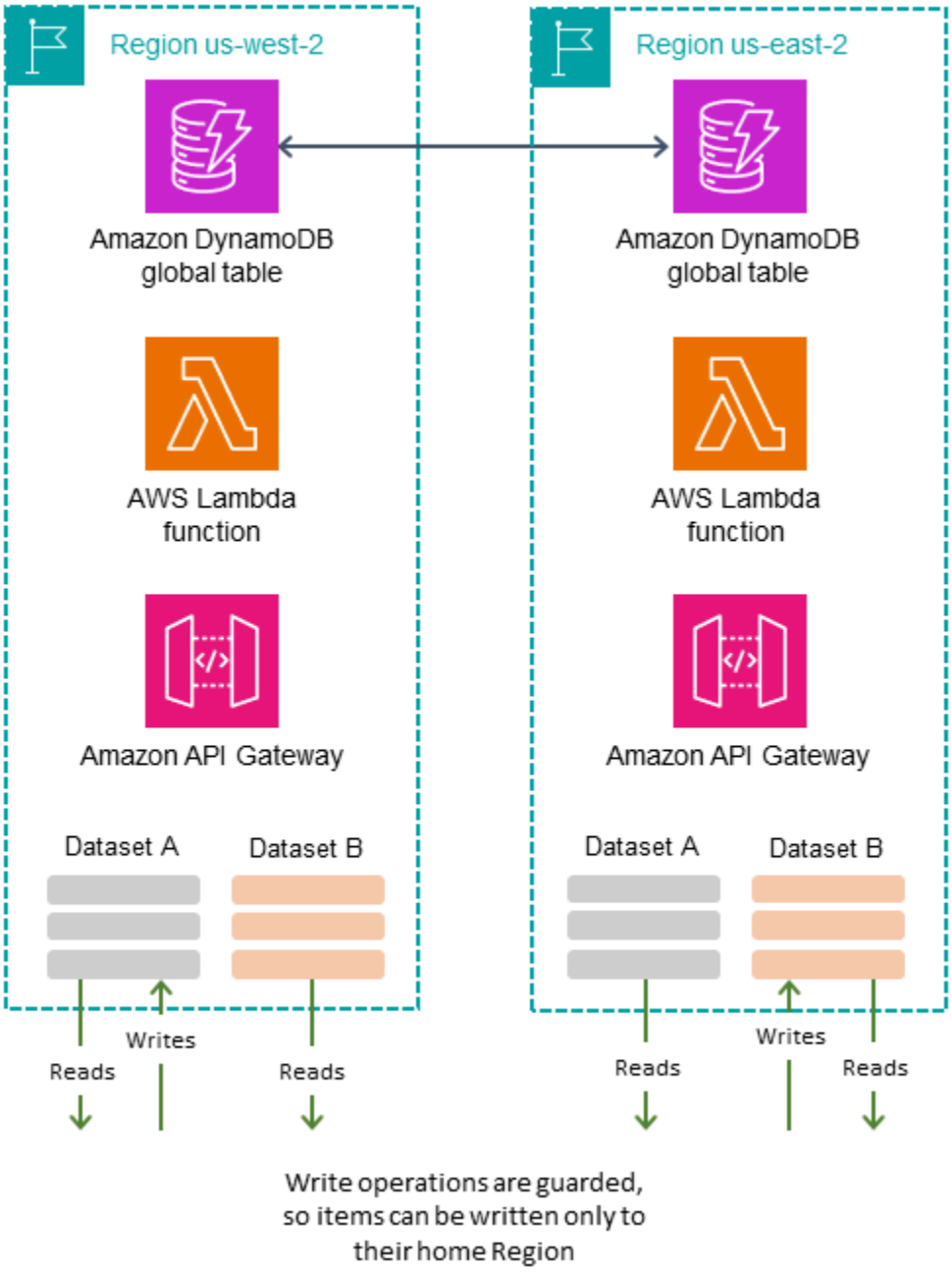
當您使用全域資料表進行低延遲、全域分佈的讀取操作時，使用寫入至一個區域模式效果良好。一個範例是大型社交媒體公司，需要在全球每個區域中提供相同的參考資料。它們不會經常更新資料，但當它們更新時，只會寫入一個區域，以避免任何潛在的寫入衝突。讀取操作一律可從任何區域進行。

另一個範例是，請考慮之前討論的實作每日現金回饋計算的金融服務公司。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追蹤現金回饋付款。如果他們想要每花費 10 美元就獎勵一分錢，他們必須Query針對前一天的所有交易，計算花費的總計，將現金回饋決策寫入新資料表，刪除查詢的項目集以將其標示為已耗用，並將它們替換為存放任何剩餘項目的單一項目，該剩餘項目應納入次日的計算。此工作需要交易，因此寫入一個區域模式時效果更佳。只要工作負載沒有重疊的機會，應用程式可以混合寫入模式，即使是在同一個資料表上。

寫入您的區域模式 (混合優先層級)

寫入至您的區域寫入模式會將不同的資料子集指派給不同的主區域，並僅允許透過其主區域對項目進行寫入操作。此模式是主動被動模式，但會根據項目指派作用中的區域。每個區域都是其自身非重疊資料集的主要區域，且寫入操作必須受到保護，以確保適當的位置。

此模式類似於寫入一個區域，但其啟用低延遲寫入操作，因為與每個使用者相關聯的資料可以放置在與該使用者更接近的網路中。它也會在區域之間更平均地分散周圍的基礎設施，並在容錯移轉情況下需要較少的工作來建置基礎設施，因為所有區域都有一部分的基礎設施已處於作用中狀態。



您可以透過多種方式判斷項目的主區域：

- 內部：資料的某些層面，例如特殊屬性或內嵌在其分割區索引鍵中的值，會使其主區域清晰。此技術在部落格文章中使用區域釘選為 Amazon DynamoDB 全域資料表中的項目設定主區域中描述。
- 交涉：每個資料集的主區域是以某些外部方式交涉，例如使用單獨的全域服務來維護指派。指派的持續時間可能有限，之後會進行重新交涉。

- 資料表導向：您可以建立與複寫區域相同的全域資料表數目，而不是建立單一複寫全域資料表。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責任。

例如，假設您正在為遊戲公司工作。您需要為全球所有玩家執行低延遲的讀取和寫入操作。您可以將每個玩家指派給最接近他們的區域。該區域會採取所有讀取和寫入操作，確保強大的read-after-write一致性。不過，當玩家出遊或他們的主區域發生中斷時，其資料的完整副本可在其他區域取得，而且玩家可以指派給不同的主區域。

另一個範例是，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。呼叫者可以使用最接近的區域，以獲得最低延遲。如果有區域中斷，使用全域資料表可讓快速復原，因為系統可以將呼叫的處理移動到已存在資料複寫副本的不同區域。

全域資料表的路由策略

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。請求會遇到該區域中的一些服務堆疊，包括可能由 AWS Lambda 函數、容器或 Amazon Elastic Compute Cloud (Amazon EC2) 節點支援的負載平衡器組成的運算層，以及其他可能的服務，包括另一個資料庫。該運算層會與 DynamoDB 通訊。它應該使用該區域的本機端點來執行此操作。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。

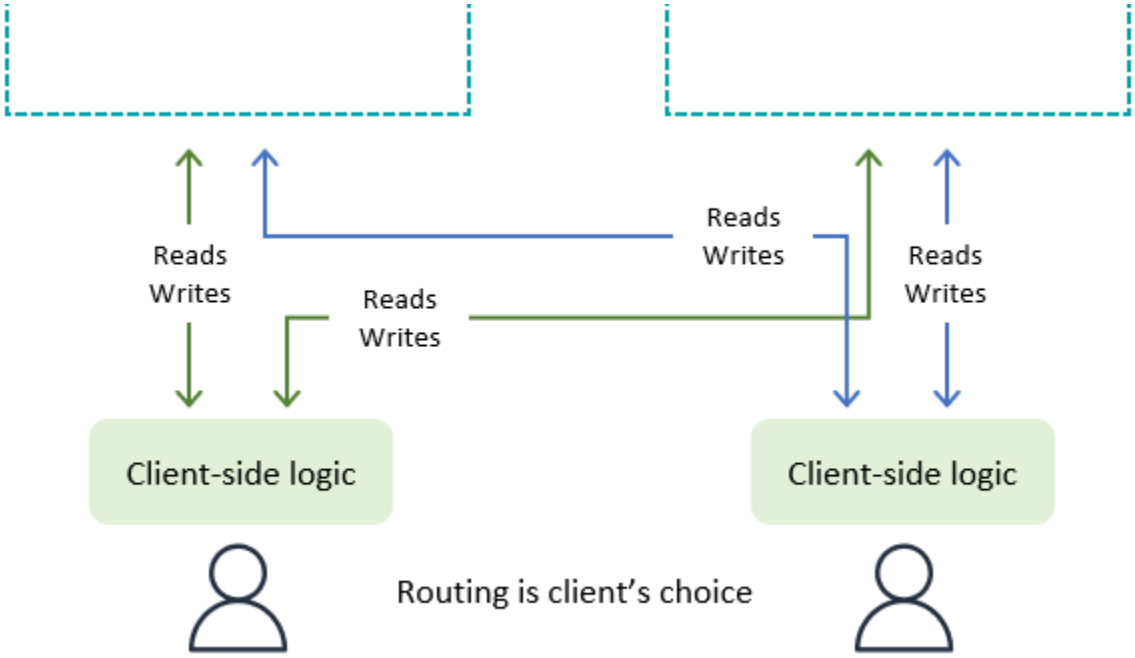
全域資料表會為不同區域中的每個堆疊提供相同資料的本機複本。如果本機 DynamoDB 資料表發生問題，您可以考慮為單一區域中的單一堆疊進行設計，並預期會對次要區域的 DynamoDB 端點進行遠端呼叫。這不是最佳實務。跨區域相關的延遲可能比本機存取高 100 倍。一系列來回的 5 個請求，本機執行時可能需要幾毫秒，但跨越全球時可能需要幾秒鐘。建議最好將終端使用者路由到另一個區域進行處理。為了確保彈性，您需要跨多個區域複寫：複寫運算層和資料層。

有許多將最終使用者請求路由到 區域進行處理的技術。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：用戶端驅動、運算層、Amazon Route 53 和 AWS Global Accelerator。

主題

用戶端驅動的請求路由

透過用戶端驅動的請求路由，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，而不是常值 DynamoDB 端點），並使用自己的內嵌邏輯來選擇要通訊的區域。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或本機執行的運作狀態檢查來選擇。



作為優勢，用戶端驅動的請求路由可以適應真實世界公有網際網路流量條件等情況，以便在注意到任何效能降低時切換區域。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，用戶端可以單邊選取其偏好的端點。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。

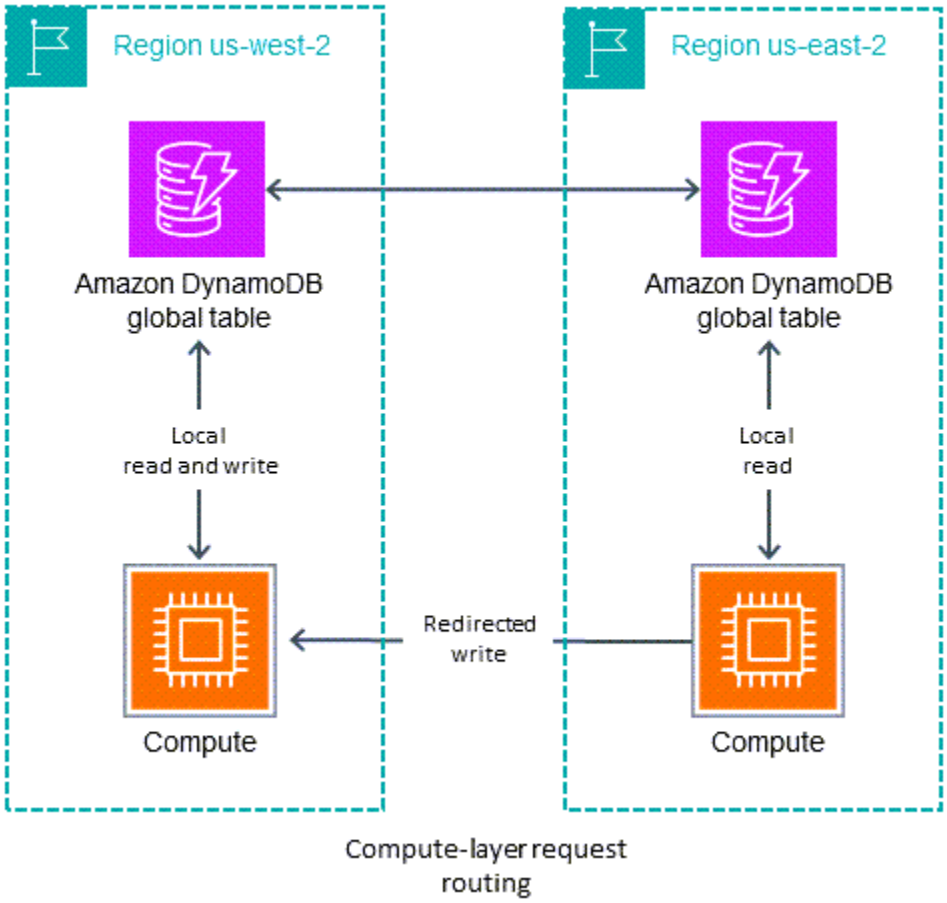
透過寫入至一個區域模式，用戶端需要一個機制，將其寫入請求路由到目前作用中的區域。這可以是基本機制，例如以經驗方式測試哪個區域目前接受寫入請求（注意任何寫入拒絕並返回替代）。或者，它可以是複雜的機制，例如使用全域協調器來查詢目前的應用程式狀態（可能建置在 Amazon Application Recovery Controller (ARC) (ARC) 路由控制上，可提供五區域、規定人數驅動的系統，以維護全域狀態，滿足這類需求）。用戶端可以決定讀取請求是否可以前往任何區域以取得最終一致性，還是必須路由至作用中區域以取得強大的一致性。

使用寫入區域模式時，用戶端需要判斷其使用之資料集的主區域。例如，如果用戶端對應至使用者帳戶，且每個使用者帳戶都歸位至區域，則用戶端可以從全域登入系統請求適當的端點指派，以搭配其登入資料使用。

例如，一家金融服務公司，透過 Web 協助使用者管理其商業財務，使用寫入您區域模式的全域資料表。每個使用者都必須登入中央服務。此服務會傳回登入資料，以及這些登入資料將運作之區域的端點。傳回的區域取決於使用者資料集目前所在的位置。憑證僅在短時間內有效。之後，網頁會自動交涉新的登入，提供將使用者活動重新引導至新區域的機會。

運算層請求路由

使用運算層請求路由時，在運算層中執行的程式碼會決定是否在本機處理請求，或將其傳遞至在另一個區域中執行的自身副本。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作用中區域，並允許本機讀取操作，同時將所有寫入操作轉送至另一個區域。此運算層程式碼必須了解資料拓撲和路由規則，並根據指定哪些資料處於作用中狀態的最新設定可靠地強制執行它們。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。

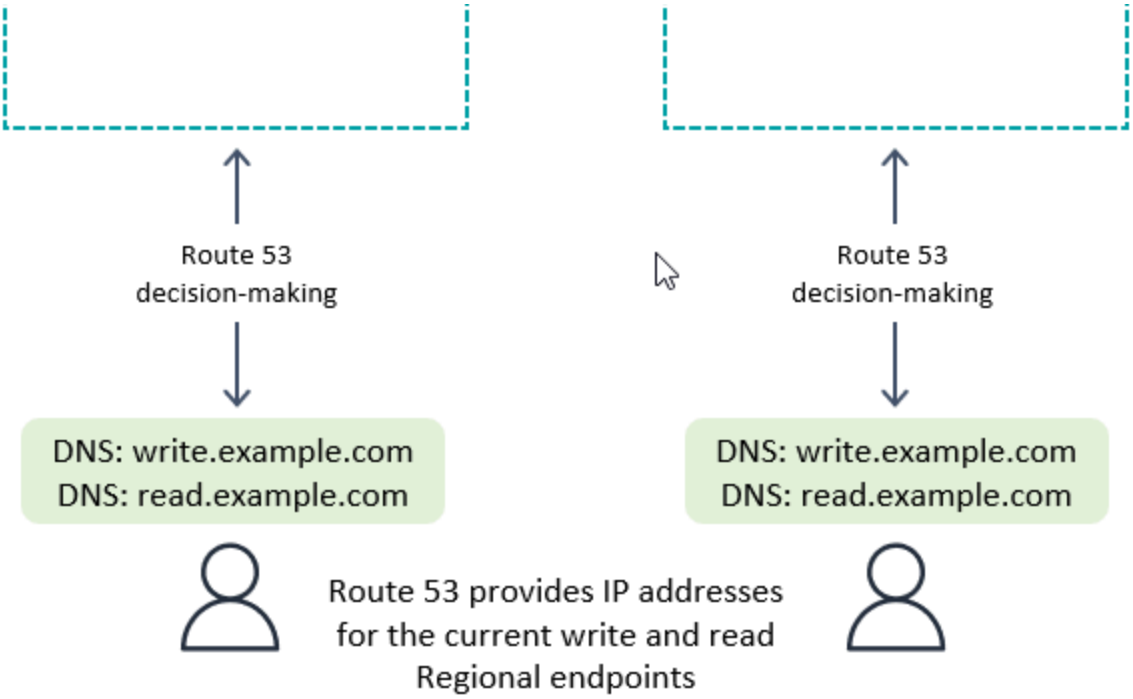


Vanguard 群組會針對此路由程序使用名為 Global Orchestration and Status Tool (GOaST) 的系統，以及名為 Global Multi-Region Library (GMRlib) 的程式庫，如 re：Invent 2022 所示。他們使用follow-the-sun主要模型。GOaST 會維護全域狀態，類似於上一節討論的 ARC 路由控制。它使用全域資料表來追蹤哪個區域是主要區域，以及排程下一個主要交換器的時間。所有讀取和寫入操作都會經過 GMRlib，其會與 GOaST 協調。GMRlib 允許在本機以低延遲執行讀取操作。對於寫入操作，GMRlib 會檢查本機區域是否為目前的主要區域。如果是，則寫入操作會直接完成。如果沒有，GMRlib 會將寫

入任務轉送至主要區域中的 GMRlib。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 Route 53 是一種網域名稱服務 (DNS) 技術。使用 Route 53 時，用戶端會查詢已知的 DNS 網域名稱來請求其端點，而 Route 53 會傳回對應至其判斷最適當的區域端點的 IP 地址 (IP)。Route 53 有一份長的路由政策清單，可用來判斷適當的區域。它也可以執行容錯移轉路由，將流量從運作狀態檢查失敗的區域路由。



透過寫入任何區域模式，或如果 與後端的運算層請求路由結合，Route 53 可以根據任何複雜的內部規則來完全自由傳回區域，例如選擇最接近網路或地理位置的區域，或任何其他選擇。

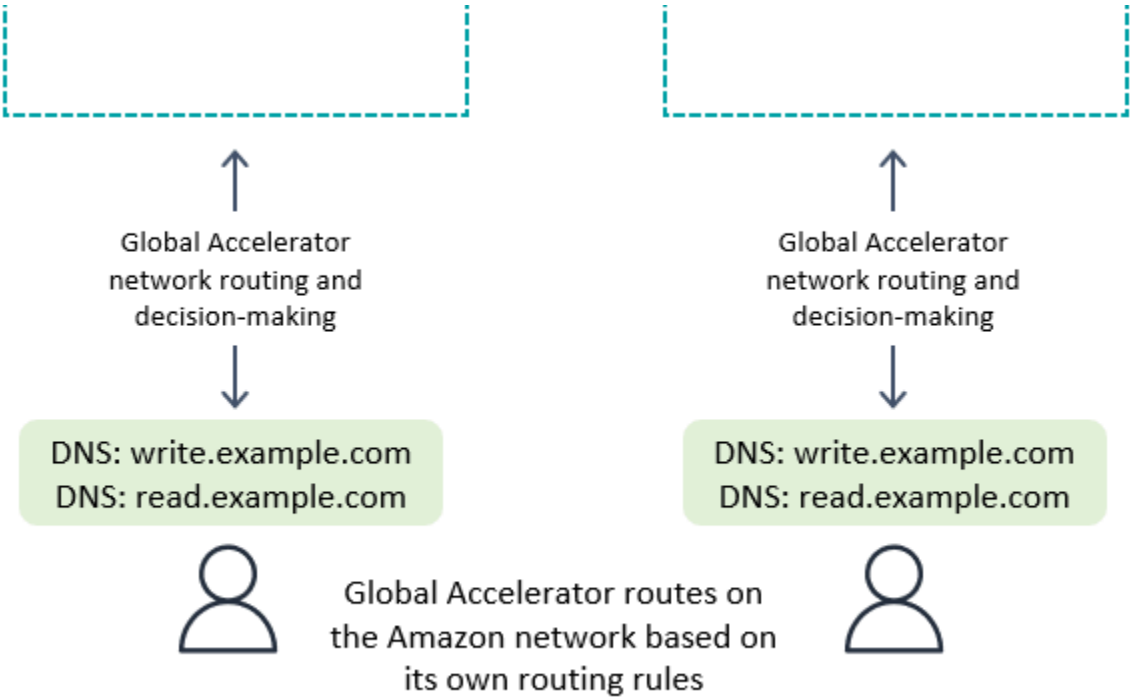
透過寫入一個區域模式，您可以設定 Route 53 以傳回目前作用中的區域 （ 使用 ARC)。如果用戶端想要連線到被動區域 （ 例如，用於讀取操作 ），可能會查詢不同的 DNS 名稱。

> **Note**
>
> 用戶端會快取 Route 53 回應中的 IP 地址一段時間，該時間由網域名稱上的存留時間 (TTL) 設定所指定的時間。較長的 TTL 會延長復原時間點目標 (RTO)，讓所有用戶端辨識新的端點。60 秒是容錯移轉使用的典型值。並非所有軟體都完全遵守 DNS TTL 過期，而且可能有多個 DNS 快取層級，例如在作業系統、虛擬機器和應用程式。

透過寫入區域模式，建議您避免 Route 53，除非您也使用運算層請求路由。

Global Accelerator 請求路由

AWS Global Accelerator 用戶端會在 Route 53 中查詢已知的網域名稱。不過，用戶端不會傳回對應至區域端點的 IP 地址，而是傳回路由至最近 AWS 節點的任何廣播靜態 IP 地址。從該邊緣位置開始，所有流量都會在私有 AWS 網路上路由到 Global Accelerator 內由路由規則所選擇的區域中的某些端點 (Network Load Balancer、Application Load Balancer、EC2 執行個體或彈性 IP 地址）。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 Global Accelerator 不依賴 DNS TTL 過期來變更路由規則，因此可以更快地調整路由。



透過寫入任何區域模式，或如果 與後端的運算層請求路由結合，Global Accelerator 可無縫運作。用戶端會連線至最近的節點，不必擔心哪個區域收到請求。

寫入一個區域模式後，Global Accelerator 路由規則必須將請求傳送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，如果請求來自任何區域，則可以使用替代 DNS 網域名稱來路由讀取請求。

透過寫入區域模式，建議您避免 Global Accelerator，除非您也使用運算層請求路由。

全域資料表的疏散程序

疏散區域是遷移活動的程序，通常是寫入活動，可能是讀取活動，遠離該區域。

疏散即時區域

您可能會因為多種原因決定撤離即時區域：作為一般商業活動的一部分（例如，如果您使用follow-the-sun、寫入一個區域模式）、因為業務決定變更目前作用中區域、回應 DynamoDB 外部軟體堆疊中的失敗，或因為您遇到一般問題，例如區域中的延遲高於一般延遲。

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，將流量路由至替代區域，並讓已疏散的區域複寫中已發生的寫入操作如往常一樣。

透過寫入至一個區域並寫入至您的區域模式，您必須確定所有對作用中區域的寫入操作都已完整記錄、串流處理和全域傳播，才能在新的作用中區域中開始寫入操作，以確保未來的寫入操作會根據資料的最新版本進行處理。

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

有特殊情況需要考慮：如果區域 A 完全離線，而不另行通知，該怎麼辦？ 這是非常不可能的，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，讀取和寫入操作可以在區域 B 中繼續，信任區域 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 恢復線上狀態為止。如果可能，例如使用等冪寫入操作，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），以填補任何潛在遺失寫入操作的差距，並讓最後一個寫入器獲勝衝突解決會抑制傳入寫入操作的最終傳播。

使用其他寫入模式，您必須考慮工作可以在稍微延時的環境下繼續進行的程度。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，假設即使區域完全中斷，您仍必須維持可用的額度餘額，而不會中斷。您可以將餘額分割為兩個不同的項目，一個位於區域 A，另一個位於區域 B，並以可用餘額的一半開始。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。

另一個範例是，假設您正在擷取 Web 表單資料。您可以使用樂觀並行控制 (OCC) 將版本指派給資料項目，並將最新版本內嵌至 Web 表單做為隱藏欄位。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。表單是第一次在 12：00 根據區域 A 建置，但 （容錯移轉後） 會嘗試寫入區域 B，並注意到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 資料庫中持有有關客戶帳戶及其財務交易的資料。如果發生完整的區域 A 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 中完全可用，或者他們希望將其帳戶隔離為已知部分，直到區域 A 恢復上線為止。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。這些帳戶會標示為隔離，直到 A 區域復原將部分資料傳播至 B 區域為止。如果 C 區域失敗，則新的 D 區域可能會輪換使用。區域 C 中的資料非常短暫，幾分鐘後，區域 D 將擁有最新的使用中寫入操作記錄，以充分發揮作用。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考量：

- 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。
- 您可能會看到不同佈建寫入容量的一個原因是由於存留時間 (TTL) 功能。在 DynamoDB 中啟用 TTL 時，您可以指定屬性名稱，其值表示項目的過期時間，以秒為單位。之後，DynamoDB 可刪除項目，而不會產生寫入成本。有了全域資料表，您可以在任何區域設定 TTL，該設定便會自動複寫到另一個區域關聯的全域資料表。當某個項目符合通過 TTL 規則刪除的條件時，該工作可以在任何區域完成。刪除操作在來源資料表上執行時不會耗用寫入單位，但複本資料表會取得該刪除操作的複寫寫入，並產生複寫寫入單位成本。
- 如果您使用自動擴展，請確定佈建的寫入容量上限設定足以處理所有寫入操作以及所有潛在的 TTL 刪除操作。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。我們建議您設定充足，以處理隨需資料表可能需要的所有寫入操作 (包括 TTL 寫入操作)。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考量：

- 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。

無論您是否使用 Route 53 來路由請求，ARC 都有準備度檢查，可用於確認 DynamoDB 區域具有類似的資料表設定和帳戶配額。這些準備度檢查也可協助您調整帳戶層級配額，使其相符。

全域資料表的準備檢查清單

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您的路由策略。
- 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 以取得營運意識。您也應該使用合成 Canary （專為偵測失敗而設計的人工請求），以及即時觀察客戶流量。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。您也可以觀察是否`ReplicationLatency`捨棄至 0，這表示複寫停滯。
- 為每個全域資料表指派充足的最大讀取和寫入設定值。
- 識別您要疏散區域的條件。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**
>
> 透過容錯移轉程序，最佳實務是僅依賴資料平面操作，而不是控制平面操作，因為某些控制平面操作可能會在區域故障期間降級。如需詳細資訊，請參閱 AWS 部落格文章使用 Amazon DynamoDB 全域資料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。
- 考慮使用 AWS Resilience Hub來評估整個應用程式的彈性 （包括全域資料表）。此服務透過其儀表板提供應用程式產品組合彈性狀態的全面檢視。
- 考慮使用 ARC 整備檢查來評估應用程式的目前組態，並追蹤最佳實務的任何偏差。
- 當您編寫運作狀態檢查以搭配 Route 53 或 Global Accelerator 使用時，請進行一組涵蓋完整資料庫流程的呼叫。如果您限制檢查以確認 DynamoDB 端點已啟動，您將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 外部堆疊中的失敗、高於平均讀取或寫入延遲等。

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 費用會在項目直接寫入或透過複寫寫入的每個區域中產生。
- 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 時間點復原 (PITR)。

如何使用 部署全域資料表 AWS CloudFormation？

- CloudFormation 將 DynamoDB 資料表和一個全域資料表顯示為兩個獨立的資源：`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 CloudFormation 中，無論複本數目多少，每個全域資料表都在單一區域中由單一堆疊控制。當您部署範本時，CloudFormation 會將所有複本建立和更新為單一堆疊操作的一部分。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。
- 如果您有一個常規資料表，並且想要將其轉換為全域資料表，同時維持 CloudFormation 受管：請將刪除政策設定為 `Retain`，從堆疊中刪除該資料表，將資料表轉換為主控台內的全域資料表，然後將全域資料表作為新的資源匯入堆疊。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 全域資料表的控制非常少，但仍需要仔細考慮。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。獎勵具有全域分佈的資料集，具有專為 99.999% 可用性設計的低延遲讀取和寫入操作。

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 Route 53 應用程式復原控制器
- ARC 整備檢查 (AWS 文件）
- Route 53 路由政策 (AWS 文件）
- AWS Global Accelerator
- DynamoDB 服務層級協議
- AWS 多區域基礎 (AWS 白皮書）
- 使用 的資料彈性設計模式 AWS(AWS re：Invent 2022 簡報）
- Fidelity 投資和 Reltio 如何使用 Amazon DynamoDB 進行現代化 (AWS re：Invent 2022 簡報）
- 多區域設計模式和最佳實務 (AWS re：Invent 2022 簡報）
- 上的災難復原 (DR) 架構 AWS，第 III 部分：指示燈和暖待命 (AWS 部落格文章）
- 使用區域釘選，為 Amazon DynamoDB 全域資料表（部落格文章）中的項目設定主區域AWS
- 監控 Amazon DynamoDB 的操作意識 (AWS 部落格文章）
- 擴展 DynamoDB：分割區、熱鍵和分割對熱影響效能的方式 (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 Global Accelerator 請求路由的端點。	2024 年 3 月 14 日
更新 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 規範性指引詞彙表

以下是 AWS Prescriptive Guidance 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 資料庫遷移至 Amazon Aurora PostgreSQL 相容版本。
- 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 資料庫遷移至 中的 Amazon Relational Database Service (Amazon RDS) for Oracle 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 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應用程式遷移至 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

請參閱屬性型存取控制。

抽象服務

請參閱 受管服務。

ACID

請參閱原子性、一致性、隔離性、耐久性。

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。它更靈活，但比主動-被動遷移需要更多的工作。

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。

彙總函數

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

AI

請參閱人工智慧。

AIOps

請參閱人工智慧操作。

匿名化

在資料集中永久刪除個人資訊的程序。匿名化有助於保護個人隱私權。匿名資料不再被視為個人資料。

反模式

經常用於重複性問題的解決方案，其中解決方案具有反生產力、無效或比替代解決方案更有效。

應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體攻擊。

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》中的 ABAC for AWS。

授權資料來源

您存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將授權資料來源中的資料複製到其他位置，以處理或修改資料，例如匿名、修訂或假名化資料。

可用區域

中的不同位置 AWS 區域 ，可隔離其他可用區域中的故障，並提供相同區域中其他可用區域的低成本、低延遲網路連線。

AWS 雲端採用架構 (AWS CAF)

的指導方針和最佳實務架構 AWS ，可協助組織制定高效且有效的計劃，以成功地移至雲端。 AWS CAF 將指導方針組織到六個重點領域：業務、人員、治理、平台、安全和營運。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。因此， AWS CAF 為人員開發、訓練和通訊提供指引，協助組織做好成功採用雲端的準備。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)

一種工具，可評估資料庫遷移工作負載、建議遷移策略，並提供工作預估值。 AWS WQF 隨附於 AWS Schema Conversion Tool (AWS SCT)。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

錯誤的機器人

旨在中斷或傷害個人或組織的機器人。

BCP

請參閱業務持續性規劃。

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 Endianness。

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

一種部署策略，您可以在其中建立兩個不同但相同的環境。您可以在一個環境 （藍色） 中執行目前的應用程式版本，並在另一個環境 （綠色） 中執行新的應用程式版本。此策略可協助您快速復原，並將影響降至最低。

機器人

透過網際網路執行自動化任務並模擬人類活動或互動的軟體應用程式。有些機器人有用或有益，例如在網際網路上為資訊編製索引的 Web 爬蟲程式。有些其他機器人稱為惡意機器人，旨在中斷或傷害個人或組織。

殭屍網路

受到惡意軟體感染且受單一方控制之機器人的網路，稱為機器人繼承器或機器人運算子。殭屍網路是擴展機器人及其影響的最佳已知機制。

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)。

碎片存取

在特殊情況下，以及透過核准的程序，讓使用者能夠快速存取他們通常無權存取 AWS 帳戶 的 。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS 實作打破玻璃程序指標。

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

CAF

請參閱AWS 雲端採用架構。

Canary 部署

版本對最終使用者的緩慢和增量版本。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 Cloud Center of Excellence。

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，以測試系統的彈性。您可以使用 AWS Fault Injection Service (AWS FIS) 執行實驗，為您的 AWS 工作負載帶來壓力，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。雲端運算通常連接到邊緣運算技術。

雲端操作模型

在 IT 組織中，用於建置、成熟和最佳化一或多個雲端環境的操作模型。如需詳細資訊，請參閱建置您的雲端操作模型。

採用雲端階段

組織在遷移至 時通常會經歷的四個階段 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

這些階段由 Stephen Orban 於部落格文章 The Journey Toward Cloud-First 和 Enterprise Strategy 部落格上的採用階段中定義。 AWS 雲端 如需有關它們如何與 AWS 遷移策略相關的詳細資訊，請參閱遷移整備指南。

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。常見的雲端儲存庫包括 GitHub或 Bitbucket Cloud。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的資料。查詢這類資料時，通常可接受慢查詢。將此資料移至效能較低且成本較低的儲存層或類別，可以降低成本。

電腦視覺 (CV)

使用機器學習從數位影像和影片等視覺化格式分析和擷取資訊的 AI 欄位。例如，Amazon SageMaker AI 提供 CV 的影像處理演算法。

組態偏離

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

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

您可以組合的 AWS Config 規則和修補動作集合，以自訂您的合規和安全檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶 和 區域中或整個組織的單一實體。如需詳細資訊，請參閱 AWS Config 文件中的一致性套件。

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 Framework 中安全支柱的元件。如需詳細資訊，請參閱資料分類。

資料偏離

生產資料與用於訓練 ML 模型的資料之間有意義的變化，或輸入資料隨時間有意義的變更。資料偏離可以降低 ML 模型預測的整體品質、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

架構架構，提供分散式、分散式資料擁有權與集中式管理。

資料最小化

僅收集和處理嚴格必要資料的原則。在 中實作資料最小化 AWS 雲端 可以降低隱私權風險、成本和分析碳足跡。

資料周邊

AWS 環境中的一組預防性防護機制，可協助確保只有信任的身分才能從預期的網路存取信任的資源。如需詳細資訊，請參閱在 上建置資料周邊 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。

資料來源

在整個資料生命週期中追蹤資料的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

支援商業智慧的資料管理系統，例如 分析。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。

深度防禦

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。當您在 上採用此策略時 AWS，您可以在 AWS Organizations 結構的不同層新增多個控制項，以協助保護資源。例如，defense-in-depth方法可能會結合多重要素驗證、網路分割和加密。

委派的管理員

在 中 AWS Organizations，相容的服務可以註冊 AWS 成員帳戶來管理組織的帳戶，並管理該服務的許可。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。

開發值串流映射 (DVSM)

一種程序，用於識別並優先考慮對軟體開發生命週期中的速度和品質造成負面影響的限制。DVSM 擴展了最初專為精簡製造實務設計的價值串流映射程序。它著重於透過軟體開發程序建立和移動價值所需的步驟和團隊。

數位分身

真實世界系統的虛擬呈現，例如建築物、工廠、工業設備或生產線。數位分身支援預測性維護、遠端監控和生產最佳化。

維度資料表

在星星結構描述中，較小的資料表包含有關事實資料表中量化資料的資料屬性。維度資料表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、篩選和結果集標記。

災難

防止工作負載或系統在其主要部署位置實現其業務目標的事件。這些事件可能是自然災難、技術故障或人為動作的結果，例如意外設定錯誤或惡意軟體攻擊。

災難復原 (DR)

您用來將災難造成的停機時間和資料遺失降至最低的策略和程序。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的 上工作負載的災難復原 AWS：雲端中的復原。

DML

請參閱資料庫處理語言。

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。

DR

請參閱災難復原。

偏離偵測

追蹤與基準組態的偏差。例如，您可以使用 AWS CloudFormation 來偵測系統資源中的偏離，也可以使用 AWS Control Tower 來偵測登陸區域中可能影響控管要求合規性的變更。 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html

DVSM

請參閱開發值串流映射。

E

EDA

請參閱探索性資料分析。

EDI

請參閱電子資料交換。

邊緣運算

提升 IoT 網路邊緣智慧型裝置運算能力的技術。與雲端運算相比，邊緣運算可以減少通訊延遲並改善回應時間。

電子資料交換 (EDI)

在組織之間自動交換商業文件。如需詳細資訊，請參閱什麼是電子資料交換。

加密

將人類可讀取的純文字資料轉換為加密文字的運算程序。

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 建立端點服務，AWS PrivateLink 並將許可授予其他 AWS 帳戶 或 AWS Identity and Access Management (IAM) 委託人。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 (ERP)

一種系統，可自動化和管理企業的關鍵業務流程 （例如會計、MES 和專案管理）。

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 () 文件中的信封加密。 AWS Key Management Service AWS KMS

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 安全概念包括身分和存取管理、偵測控制、基礎設施安全、資料保護和事件回應。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

事實資料表

星狀結構描述中的中央資料表。它存放有關業務操作的量化資料。一般而言，事實資料表包含兩種類型的資料欄：包含度量的資料，以及包含維度資料表外部索引鍵的資料欄。

快速失敗

一種使用頻繁且增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離界限

在 中 AWS 雲端，像是可用區域 AWS 區域、控制平面或資料平面等界限會限制故障的影響，並有助於改善工作負載的彈性。如需詳細資訊，請參閱AWS 故障隔離界限。

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 的機器學習模型可解譯性 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。

少量擷取提示

在要求 LLM 執行類似的任務之前，提供少量示範任務和所需輸出的範例。此技術是內容內學習的應用程式，其中模型會從內嵌在提示中的範例 (快照) 中學習。少量的提示對於需要特定格式、推理或網域知識的任務來說非常有效。另請參閱零鏡頭提示。

FGAC

請參閱精細存取控制。

精細存取控制 (FGAC)

使用多個條件來允許或拒絕存取請求。

閃切遷移

一種資料庫遷移方法，透過變更資料擷取使用連續資料複寫，以盡可能在最短的時間內遷移資料，而不是使用分階段方法。目標是將停機時間降至最低。

FM

請參閱基礎模型。

基礎模型 (FM)

大型深度學習神經網路，已在廣義和未標記資料的大量資料集上進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及以自然語言交談。如需詳細資訊，請參閱什麼是基礎模型。

G

生成式 AI

已針對大量資料進行訓練的 AI 模型子集，可使用簡單的文字提示建立新的內容和成品，例如影像、影片、文字和音訊。如需詳細資訊，請參閱什麼是生成式 AI。

地理封鎖

請參閱地理限制。

地理限制 (地理封鎖)

Amazon CloudFront 中的選項，可防止特定國家/地區的使用者存取內容分發。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 工作流程會被視為舊版，而以幹線為基礎的工作流程是現代、偏好的方法。

黃金影像

系統或軟體的快照，做為部署該系統或軟體新執行個體的範本。例如，在製造中，黃金映像可用於在多個裝置上佈建軟體，並有助於提高裝置製造操作的速度、可擴展性和生產力。

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。它們是透過使用 AWS Config AWS Security Hub、Amazon GuardDuty、、Amazon Inspector AWS Trusted Advisor和自訂 AWS Lambda 檢查來實作。

H

HA

請參閱高可用性。

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)

在遇到挑戰或災難時，工作負載能夠在不介入的情況下持續運作。HA 系統的設計目的是自動容錯移轉、持續提供高品質的效能，並處理不同的負載和故障，並將效能影響降至最低。

歷史現代化

一種方法，用於現代化和升級操作技術 (OT) 系統，以更好地滿足製造業的需求。歷史資料是一種資料庫，用於從工廠中的各種來源收集和存放資料。

保留資料

從用於訓練機器學習模型的資料集中保留的部分歷史標記資料。您可以使用保留資料，透過比較模型預測與保留資料來評估模型效能。

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。

熱資料

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，通常會在典型 DevOps 發行工作流程之外執行修補程式。

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

將基礎設施視為程式碼。

身分型政策

連接至一或多個 IAM 主體的政策，可定義其在 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。

不可變的基礎設施

為生產工作負載部署新基礎設施的模型，而不是更新、修補或修改現有的基礎設施。不可變基礎設施本質上比可變基礎設施更一致、可靠且可預測。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的使用不可變基礎設施部署最佳實務。

傳入 (輸入) VPC

在 AWS 多帳戶架構中，接受、檢查和路由來自應用程式外部之網路連線的 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。

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

2016 年 Klaus Schwab 推出的術語，透過連線能力、即時資料、自動化、分析和 AI/ML 的進展，指製造程序的現代化。

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 VPCs 之間 （在相同或不同的 中 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 的機器學習模型可解譯性 AWS。

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

標籤型存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中使用者和資料本身都會獲得明確指派的安全標籤值。使用者安全標籤和資料安全標籤之間的交集會決定使用者可以看到哪些資料列和資料欄。

登陸區域

登陸區域是架構良好的多帳戶 AWS 環境，可擴展且安全。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。

大型語言模型 (LLM)

預先訓練大量資料的深度學習 AI 模型。LLM 可以執行多個任務，例如回答問題、摘要文件、將文字翻譯成其他語言，以及完成句子。如需詳細資訊，請參閱什麼是 LLMs。

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱標籤型存取控制。

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

請參閱 7 個 R。

小端序系統

首先儲存最低有效位元組的系統。另請參閱 Endianness。

LLM

請參閱大型語言模型。

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

旨在危及電腦安全或隱私權的軟體。惡意軟體可能會中斷電腦系統、洩露敏感資訊，或取得未經授權的存取。惡意軟體的範例包括病毒、蠕蟲、勒索軟體、特洛伊木馬、間諜軟體和鍵盤記錄器。

受管服務

AWS 服務 會 AWS 操作基礎設施層、作業系統和平台，而您會存取端點來存放和擷取資料。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 (MES)

一種軟體系統，用於追蹤、監控、記錄和控制生產程序，將原物料轉換為現場成品。

MAP

請參閱遷移加速計劃。

機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在操作時強化和改善自身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的建置機制。

成員帳戶

除了屬於組織一部分的管理帳戶 AWS 帳戶 之外的所有 AWS Organizations。一個帳戶一次只能是一個組織的成員。

製造執行系統

請參閱製造執行系統。

訊息佇列遙測傳輸 (MQTT)

根據發佈/訂閱模式的輕量型machine-to-machine(M2M) 通訊協定，適用於資源受限的 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，請參閱使用無 AWS 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，請參閱在 上實作微服務 AWS。

Migration Acceleration Program (MAP)

一種 AWS 計畫，提供諮詢支援、訓練和服務，協助組織建立強大的營運基礎，以移至雲端，並協助抵銷遷移的初始成本。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。遷移工廠團隊通常包括營運、業務分析師和擁有者、遷移工程師、開發人員以及從事 Sprint 工作的 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。遷移中繼資料的範例包括目標子網路、安全群組和 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：使用 AWS Application Migration Service 重新託管遷移至 Amazon EC2。

遷移組合評定 (MPA)

線上工具，提供驗證商業案例以遷移至 的資訊 AWS 雲端。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)。MPA 工具 （需要登入） 可供所有 AWS 顧問和 APN 合作夥伴顧問免費使用。

遷移準備程度評定 (MRA)

使用 AWS CAF 取得組織雲端整備狀態的洞見、識別優缺點，以及建立行動計劃以消除已識別差距的程序。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

用來將工作負載遷移至 的方法 AWS 雲端。如需詳細資訊，請參閱此詞彙表中的 7 個 Rs 項目，並請參閱動員您的組織以加速大規模遷移。

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《》中的現代化應用程式的策略 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《》中的評估應用程式的現代化準備 AWS 雲端程度。

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

請參閱遷移產品組合評估。

MQTT

請參閱訊息佇列遙測傳輸。

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？」

可變基礎設施

更新和修改生產工作負載現有基礎設施的模型。為了提高一致性、可靠性和可預測性， AWS Well-Architected Framework 建議使用不可變基礎設施做為最佳實務。

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 操作整合。

OLA

請參閱操作層級協議。

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。

OPC-UA

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 提供資料加密、身分驗證和授權機制的互通性標準。

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)。

操作整備審查 (ORR)

問題和相關最佳實務的檢查清單，可協助您了解、評估、預防或減少事件和可能失敗的範圍。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的操作準備度審查 (ORR)。

操作技術 (OT)

使用實體環境控制工業操作、設備和基礎設施的硬體和軟體系統。在製造業中，整合 OT 和資訊技術 (IT) 系統是工業 4.0 轉型的關鍵重點。

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 建立的線索 AWS CloudTrail 會記錄 AWS 帳戶 組織中所有 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 文件中的建立組織追蹤。

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 遷移策略中，此架構稱為人員加速，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)

CloudFront 中的增強型選項，用於限制存取以保護 Amazon Simple Storage Service (Amazon S3) 內容。OAC 支援所有 S3 儲存貯體中的所有伺服器端加密 AWS KMS (SSE-KMS) AWS 區域，以及對 S3 儲存貯體的動態PUT和DELETE請求。

原始存取身分 (OAI)

CloudFront 中的一個選項，用於限制存取以保護 Amazon S3 內容。當您使用 OAI 時，CloudFront 會建立一個可供 Amazon S3 進行驗證的主體。經驗證的主體只能透過特定 CloudFront 分發來存取 S3 儲存貯體中的內容。另請參閱 OAC，它可提供更精細且增強的存取控制。

ORR

請參閱操作整備審核。

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之網路連線的 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。

個人身分識別資訊 (PII)

直接檢視或與其他相關資料配對時，可用來合理推斷個人身分的資訊。PII 的範例包括名稱、地址和聯絡資訊。

PII

請參閱個人身分識別資訊。

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。

PLC

請參閱可程式設計邏輯控制器。

PLM

請參閱產品生命週期管理。

政策

可定義許可的物件（請參閱身分型政策）、指定存取條件（請參閱資源型政策），或定義組織中所有帳戶的最大許可 AWS Organizations（請參閱服務控制政策）。

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 `WHERE`子句中。

述詞下推

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

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

設計隱私權

透過整個開發程序將隱私權納入考量的系統工程方法。

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

旨在防止部署不合規資源的安全控制。這些控制項會在佈建資源之前對其進行掃描。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的控制項參考指南，並參閱實作安全控制項中的主動控制項。 AWS

產品生命週期管理 (PLM)

管理產品整個生命週期的資料和程序，從設計、開發和啟動，到成長和成熟，再到拒絕和移除。

生產環境

請參閱 環境。

可程式邏輯控制器 (PLC)

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

提示鏈結

使用一個 LLM 提示的輸出做為下一個提示的輸入，以產生更好的回應。此技術用於將複雜任務分解為子任務，或反覆精簡或展開初步回應。它有助於提高模型回應的準確性和相關性，並允許更精細、個人化的結果。

擬匿名化

將資料集中的個人識別符取代為預留位置值的程序。假名化有助於保護個人隱私權。假名化資料仍被視為個人資料。

發佈/訂閱 (pub/sub)

一種模式，可啟用微服務之間的非同步通訊，以提高可擴展性和回應能力。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頻道。系統可以新增新的微服務，而無需變更發佈服務。

Q

查詢計劃

一系列步驟，如指示，用於存取 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

RACI 矩陣

請參閱負責、負責、諮詢、告知 (RACI)。

RAG

請參閱擷取增強生成。

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。

RASCI 矩陣

請參閱負責、負責、諮詢、告知 (RACI)。

RCAC

請參閱資料列和資料欄存取控制。

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。

重新架構師

請參閱 7 Rs。

復原點目標 (RPO)

自上次資料復原點以來可接受的時間上限。這會決定最後一個復原點與服務中斷之間可接受的資料遺失。

復原時間目標 (RTO)

服務中斷和服務還原之間的可接受延遲上限。

重構

請參閱 7 個 R。

區域

地理區域中的 AWS 資源集合。每個 AWS 區域 都獨立於其他 ，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 您的帳戶可以使用哪些。

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。

重新託管

請參閱 7 個 R。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新放置

請參閱 7 Rs。

Replatform

請參閱 7 Rs。

回購

請參閱 7 Rs。

彈性

應用程式抵禦中斷或從中斷中復原的能力。在 中規劃彈性時，高可用性和災難復原是常見的考量 AWS 雲端。如需詳細資訊，請參閱AWS 雲端 彈性。

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

定義所有涉及遷移活動和雲端操作之各方的角色和責任的矩陣。矩陣名稱衍生自矩陣中定義的責任類型：負責人 (R)、責任 (A)、已諮詢 (C) 和知情 (I)。支援 (S) 類型為選用。如果您包含支援，則矩陣稱為 RASCI 矩陣，如果您排除它，則稱為 RACI 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

請參閱 7 Rs。

淘汰

請參閱 7 個 R。

檢索增強生成 (RAG)

一種生成式 AI 技術，其中 LLM 會在產生回應之前參考訓練資料來源以外的授權資料來源。例如，RAG 模型可能會對組織的知識庫或自訂資料執行語意搜尋。如需詳細資訊，請參閱什麼是 RAG。

輪換

定期更新秘密的程序，讓攻擊者更難存取登入資料。

資料列和資料欄存取控制 (RCAC)

使用已定義存取規則的基本彈性 SQL 表達式。RCAC 包含資料列許可和資料欄遮罩。

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

許多身分提供者 (IdP) 使用的開放標準。此功能可啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS Management Console 或呼叫 AWS API 操作，而無需為您組織中的每個人在 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。

SCADA

請參閱監督控制和資料擷取。

SCP

請參閱服務控制政策。

秘密

您以加密形式存放的 AWS Secrets Manager機密或限制資訊，例如密碼或使用者登入資料。它由秘密值及其中繼資料組成。秘密值可以是二進位、單一字串或多個字串。如需詳細資訊，請參閱 Secrets Manager 文件中的 Secrets Manager 秘密中的什麼內容？。

設計安全性

透過整個開發程序將安全性納入考量的系統工程方法。

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。安全控制有四種主要類型：預防性、偵測性、回應性和主動性。

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

預先定義和程式設計的動作，旨在自動回應或修復安全事件。這些自動化可做為偵測或回應式安全控制，協助您實作 AWS 安全最佳實務。自動化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

由 AWS 服務 接收資料的 在其目的地加密資料。

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 文件中的服務控制政策。

服務端點

的進入點 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。

服務層級指標 (SLI)

服務效能層面的測量，例如其錯誤率、可用性或輸送量。

服務層級目標 (SLO)

代表服務運作狀態的目標指標，由服務層級指標測量。

共同責任模式

描述您與 共同 AWS 承擔雲端安全與合規責任的模型。 AWS 負責雲端的安全，而 負責雲端的安全。如需詳細資訊，請參閱共同責任模式。

SIEM

請參閱安全資訊和事件管理系統。

單一故障點 (SPOF)

應用程式的單一關鍵元件故障，可能會中斷系統。

SLA

請參閱服務層級協議。

SLI

請參閱服務層級指標。

SLO

請參閱服務層級目標。

先拆分後播種模型

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 中的階段式應用程式現代化方法 AWS 雲端。

SPOF

請參閱單一故障點。

星狀結構描述

使用一個大型事實資料表來存放交易或測量資料的資料庫組織結構，並使用一或多個較小的維度資料表來存放資料屬性。此結構旨在用於資料倉儲或商業智慧用途。

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。

監控控制和資料擷取 (SCADA)

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

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

以模擬使用者互動的方式測試系統，以偵測潛在問題或監控效能。您可以使用 Amazon CloudWatch Synthetics 來建立這些測試。

系統提示

一種向 LLM 提供內容、指示或指導方針以指示其行為的技術。系統提示有助於設定內容，並建立與使用者互動的規則。

T

標籤

做為中繼資料以組織 AWS 資源的鍵值對。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 文件中的什麼是傳輸閘道。

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

將許可授予您指定的服務，以代表您在組織中 AWS Organizations 及其帳戶中執行任務。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 文件中的 AWS Organizations 搭配使用 AWS Organizations 與其他 AWS 服務。

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

兩個比薩就能吃飽的小型 DevOps 團隊。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。

未區分的任務

也稱為繁重工作，這是建立和操作應用程式的必要工作，但不為最終使用者提供直接價值或提供競爭優勢。未區分任務的範例包括採購、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

危害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。

暖資料

不常存取的資料。查詢這類資料時，通常可接受中等速度的查詢。

視窗函數

SQL 函數，對與目前記錄有某種程度關聯的資料列群組執行計算。視窗函數適用於處理任務，例如根據目前資料列的相對位置計算移動平均值或存取資料列的值。

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。

WORM

請參閱寫入一次，讀取許多。

WQF

請參閱AWS 工作負載資格架構。

寫入一次，讀取許多 (WORM)

儲存模型，可一次性寫入資料，並防止刪除或修改資料。授權使用者可以視需要多次讀取資料，但無法變更資料。此資料儲存基礎設施被視為不可變。

Z

零時差入侵

利用零時差漏洞的攻擊，通常是惡意軟體。

零時差漏洞

生產系統中未緩解的缺陷或漏洞。威脅行為者可以使用這種類型的漏洞來攻擊系統。開發人員經常因為攻擊而意識到漏洞。

零鏡頭提示

提供 LLM 執行任務的指示，但沒有可協助引導任務的範例 (快照)。LLM 必須使用其預先訓練的知識來處理任務。零鏡頭提示的有效性取決於任務的複雜性和提示的品質。另請參閱少量擷取提示。

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。