AWS 多區域基本概念

AWS 方案指引

AWS 方案指引: AWS 多區域基本概念

Table of Contents

AWS 多區域基本概念

John Formento，Amazon Web Services (AWS)

2024 年 12 月 (文件歷史記錄)

此進階的 300 級指南適用於在 上建置工作負載的雲端架構師和資深領導者， AWS 並有興趣使用多區域架構來改善工作負載的彈性。本指南假設對基礎設施和服務有 AWS 基準知識。它概述了常見的多區域使用案例，分享了有關設計、開發和部署的基本多區域概念和影響，並提供規範性指導，協助您更好地判斷多區域架構是否適合您的工作負載。

您是 Well-Architected 嗎？

AWS Well-Architected Framework 可協助您了解在雲端建置系統時所做決策的優缺點。架構的六個支柱提供架構最佳實務，用於設計和操作可靠、安全、有效率、經濟實惠且永續的系統。您可以使用 AWS Well-Architected Tool免費提供的 AWS Management Console，透過回答每個支柱的一組問題來檢閱工作負載是否符合這些最佳實務。

如需雲端架構的其他專家指引和最佳實務，包括參考架構部署、圖表和技術指南，請參閱 AWS 架構中心。

簡介

每個 AWS 區域都包含一個地理區域內多個獨立且實體分隔的可用區域。維護每個區域中軟體服務之間的嚴格邏輯分隔。這種有目的的設計可確保某個區域中的基礎設施或服務故障不會導致另一個區域中的相互關聯故障。

大多數 AWS 使用者可以使用多個可用區域或區域，在單一區域中實現工作負載的彈性目標 AWS 服務。不過，一部分使用者追求多區域架構有三個原因：

- 他們對於其最高層工作負載具有高可用性和操作連續性需求，並希望從影響單一區域中資源的損害建立邊界復原時間。
- 他們需要滿足要求工作負載在特定司法管轄區內操作的資料主權要求（例如遵守當地法律、法規和合規）。
- 他們需要在最接近最終使用者的位置執行工作負載，以改善工作負載的效能和客戶體驗。

本指南著重於高可用性和營運要求的持續性，並協助您了解為工作負載採用多區域架構的考量事項。它說明了適用於多區域工作負載的設計、開發和部署的基本概念，並提供規範性架構，協助您判斷多區

域架構是否適合特定工作負載。您需要確保多區域架構是工作負載的正確選擇，因為這些架構具有挑戰性，而且如果多區域架構未正確建置，工作負載的整體可用性可能會降低。

單一區域中的彈性工程和操作

在深入探討多區域概念之前，請先確認您的工作負載已在單一區域中盡可能具有彈性。若要達成此目的，請根據 AWS Well-Architected Framework 的可靠性支柱和卓越營運支柱來評估工作負載，並根據權衡和風險評估進行任何必要的變更。 AWS Well-Architected 架構涵蓋下列概念：

- 以網域邊界為基礎的工作負載分割
- 明確定義的服務合約
- 相依性管理和耦合
- 處理失敗、重試和退避策略
- 冪等操作和有狀態與無狀態交易
- 操作準備和變更管理
- 了解工作負載運作狀態
- 回應事件

若要進一步採取單一區域彈性，請檢閱並套用 paperAdvanced Multi-AZ Resilience Patterns：Detecting and Mitigating Gray Failures 中討論的概念。本白皮書提供在每個可用區域中使用複本的最佳實務，以包含故障並擴展 AWS Well Architected Framework 中介紹的多可用區域概念。雖然多區域架構可以緩解與可用區域繫結的故障模式，但您應該考慮使用多區域方法來權衡。這就是為什麼我們建議您從多可用區方法開始，然後針對多區域架構的基本原理評估特定工作負載，以判斷多區域方法是否可以提高工作負載的彈性。

多區域基本知識 1：了解需求

如前所述，高可用性和操作持續性是追求多區域架構的常見原因。可用性指標會測量工作負載在定義期間內可供使用的時間百分比，而操作指標的持續性則會測量大規模且通常更長持續時間事件的復原時間。

測量可用性是幾乎持續的過程。特定測量可能會有所不同，但通常會圍繞目標可用性指標進行合併，通常稱為九個（例如 99.99% 的可用性）。使用可用性目標時，一個大小並不符合全部。您應該在工作負載層級建立可用性目標，並將非關鍵元件與關鍵元件分開，而不是在所有工作負載中套用單一目標。

為了持續操作，通常會使用下列point-in-time測量：

- 復原時間目標 (RTO) – RTO 是服務中斷和服務還原之間的最大可接受延遲。此值會決定服務受損的可接受持續時間。
- 復原點目標 (RPO) – RPO 是自上次資料復原點以來可接受的時間上限。這會決定在最新復原點和服務中斷之間，哪些資料被視為可接受的資料遺失。

與設定可用性目標類似，RTO 和 RPO 也應在工作負載層級定義。更積極的營運持續性或高可用性需要增加投資。也就是說，並非所有應用程式都可以要求或需要相同等級的彈性。調整業務和 IT 擁有者，根據業務影響來評估應用程式的重要性，然後據此進行分層，有助於提供起點。下表提供分層的範例。

此資料表顯示服務層級協議 (SLAs) 彈性分層的範例。

彈性方案	可用性 SLA	可接受的停機時間/年
白金	99.99%	52.60 分鐘
黃金	99.90%	8.77 小時
銀卡	99.5%	1.83 天

下表顯示 RTO 和 RPO 彈性分層的範例。

彈性方案	最大 RTO	最大 RPO	條件	成本
白金	15 分鐘	5 分鐘	關鍵任務工作負載	$$$
黃金	15 分鐘 – 6 小時	2 小時	重要但不重要的任務關鍵工作負載	$$
銀卡	6 小時 – 幾天	24 小時	非關鍵工作負載	$

當您設計彈性工作負載時，請考慮高可用性與營運持續性之間的關係。例如，如果工作負載需要 99.99% 的可用性，則每年不超過 53 分鐘的停機時間是可容忍的。至少需要 5 分鐘才能偵測到故障，操作員可能需要 10 分鐘才能參與、對復原步驟做出決策，並執行這些步驟。從單一問題中復原需要 30 到 45 分鐘並不罕見。在這種情況下，具有多區域策略以提供隔離的執行個體，以消除相互關聯的影響是有益的。這可讓您在限定時間內容錯移轉，同時獨立分類初始受損，以允許持續操作。這是定義適當週框復原時間並確保需要一致性的地方。

多區域方法可能適用於具有極端可用性需求（例如 99.99% 或更高可用性）的任務關鍵工作負載，或只能透過容錯移轉至另一個區域來滿足的嚴格連續性操作需求。不過，這些要求通常僅適用於企業工作負載產品組合的一小部分，其具有以分鐘或小時為單位的週框復原時間。除非應用程式需要幾分鐘或幾個小時的復原時間，否則最好等待應用程式的區域中斷在受影響的區域內進行修復。這種方法通常與低階工作負載保持一致。

在實作多區域架構之前，業務決策者和技術團隊應符合成本影響，包括營運和基礎設施成本驅動因素。典型的多區域架構會產生的成本是單一區域方法的兩倍。雖然業務持續性有數個多區域模式，例如使用熱待命、暖待命或指示燈執行，但達到復原目標風險最低的模式將涉及執行熱待命，並將工作負載的成本加倍。

關鍵指引

- 每個工作負載都應建立 RTO 和 RPO 等營運目標的可用性和持續性，並與業務和 IT 利益相關者保持一致。
- 可在單一區域內達成大多數的營運目標可用性和持續性。對於無法在單一區域內實現的目標，請考慮在成本、複雜性和利益之間的權衡方面有明確的檢視的多區域。

多區域基礎 2：了解資料

當您採用多區域架構時，管理資料是一個非小問題。區域之間的地理距離會強加不可避免的延遲，其表現為跨區域複寫資料所需的時間。必須權衡可用性、資料一致性，以及為使用多區域架構的工作負載引入更高的延遲。無論您是使用非同步還是同步複寫，您都需要修改應用程式來處理複寫技術強加的行為變更。資料一致性和延遲的挑戰使得採用專為單一區域設計的現有應用程式變得非常困難，並使其成為多區域應用程式。了解特定工作負載的資料一致性要求和資料存取模式對於權衡權衡至關重要。

2.a：了解資料一致性要求

CAP 理論提供有關資料一致性、可用性和網路分割區之間權衡的推理參考。工作負載只能同時滿足其中兩個需求。根據定義，多區域架構包含區域之間的網路分割區，因此您必須在可用性和一致性之間進行選擇。

如果您選取跨區域資料的可用性，則在交易寫入操作期間不會產生顯著的延遲，因為在複寫完成之前，對跨區域之間遞交資料非同步複寫的依賴會導致跨區域的一致性降低。使用非同步複寫時，當主要區域發生故障時，寫入操作很可能會等待主要區域的複寫。這會導致在複寫恢復之前無法使用最新資料的案例，並且需要調節程序來處理未從發生中斷的區域複寫的傳輸中交易。此案例需要了解您的商業邏輯，並建立特定程序來重播交易或在區域之間比較資料存放區。

對於偏好非同步複寫的工作負載，您可以使用 服務，例如 Amazon AuroraandAmazon DynamoDB 進行非同步跨區域複寫。Amazon Aurora 全域資料庫和Amazon DynamoDB 全域資料表都有預設的Amazon CloudWatchmetrics，以協助監控複寫延遲。Aurora 全域資料庫包含一個寫入資料的主要區域，以及最多五個唯讀次要區域。DynamoDB 全域資料表包含跨資料寫入和讀取之任何數量區域的多重作用中複本資料表。

設計工作負載以利用事件驅動型架構是多區域策略的優勢，因為它意味著工作負載可以接受資料的非同步複寫，並透過重新播放事件來啟用狀態重建。由於串流和簡訊服務緩衝訊息承載資料位於單一區域，因此區域容錯移轉或容錯回復程序必須包含重新導向用戶端輸入資料流程的機制。此程序也必須協調存放在發生中斷的區域中的傳輸中或未交付承載。

如果您選擇 CAP 一致性要求，並使用跨 區域的同步複寫資料庫來支援同時從多個 區域執行的應用程式，則可以消除資料遺失的風險，並在區域之間保持資料同步。不過，這帶來了更高的延遲特性，因為寫入需要遞交到多個區域，而且區域彼此之間可以是數百或數千英里。您需要在應用程式設計中考慮此延遲特性。此外，同步複寫可能會帶來相互關聯的失敗機會，因為寫入需要遞交至多個區域才能成功。如果一個區域內有損害，您將需要形成規定人數，才能成功寫入。這通常包括在三個區域中設定資料庫，並建立三個區域中的兩個仲裁。Paxos 等技術可協助同步複寫和遞交資料，但需要大量的開發人員投資。

當寫入涉及跨多個區域的同步複寫以滿足強大的一致性要求時，寫入延遲會增加一個數量級。較高的寫入延遲並非您通常可以在沒有重大變更的情況下修改為應用程式，例如重新檢視應用程式的逾時和重試策略。理想情況下，在第一次設計應用程式時，必須將其納入考量。對於同步複寫為優先順序的多區域工作負載， AWS Partner 解決方案可以提供協助。

2.b：了解資料存取模式

工作負載資料存取模式是讀取密集型或寫入密集型。了解特定工作負載的此特性可協助您選取適當的多區域架構。

對於完全唯讀的讀取密集型工作負載，例如靜態內容，您可以實現主動-主動多區域架構，與寫入密集型工作負載相比，其工程複雜性較低。使用內容交付網路 (CDN) 在邊緣提供靜態內容，透過快取最接近最終使用者的內容來確保可用性；使用 Amazon CloudFront 內的原始伺服器容錯移轉等功能集有助於實現此目標。另一個選項是在多個區域中部署無狀態運算，並使用 DNS 將使用者路由到最近的區域來讀取內容。您可以使用 Amazon Route 53 搭配地理位置路由政策來達成此目標。

對於讀取流量百分比大於寫入流量的讀取密集型工作負載，您可以使用 aread local， write global strategy。這表示所有寫入請求都會傳送至特定區域的資料庫，資料會以非同步方式複寫至所有其他區域，而且讀取可以在任何區域中完成。這種方法需要工作負載來接受最終一致性，因為本機讀取可能會因為跨區域寫入複寫的延遲增加而變得過時。

Aurora 全域資料庫可協助在待命區域中佈建讀取複本，該複本只能在本機處理所有讀取流量，並在特定區域中佈建單一主要資料存放區來處理寫入流量。資料會從主要資料庫非同步複寫至待命資料庫（僅供讀取複本），而且如果您需要將操作容錯移轉至待命區域，則可以將待命資料庫提升為主要資料庫。您也可以在此方法中使用 DynamoDB。DynamoDB 全域資料表可以跨區域佈建複本資料表，每個區域都可以擴展以支援任何數量的本機讀取或寫入流量。當應用程式將資料寫入某個區域的複本列表時，DynamoDB 會自動將寫入傳播到另一個 區域的其他複本列表。使用此組態，資料會從定義的主要區域非同步複寫到待命區域中的複本資料表。任何區域中的複本資料表一律可以接受寫入，因此在應用程式層級管理將待命區域提升為主要區域。同樣地，工作負載必須接受最終一致性，如果一開始就不是為此設計，則可能需要重寫。

對於寫入密集型工作負載，應選取主要區域，且應將容錯移轉至待命區域的功能設計為工作負載。與主動-主動方法相比，主要-待命方法具有額外的權衡。這是因為對於主動-主動架構，必須重寫工作負載以處理智慧路由到區域、建立工作階段親和性、確保等冪交易，以及處理潛在的衝突。

大多數使用多區域方法來恢復的工作負載不需要主動-主動方法。您可以使用碎片策略，透過限制整個用戶端基礎的損害影響範圍來提供更高的彈性。如果您可以有效地碎片化用戶端基礎，您可以為每個碎片選取不同的主要區域。例如，您可以碎片用戶端，讓一半的用戶端符合區域 1，一半符合區域 2。透

過將區域視為儲存格，您可以建立多區域儲存格方法，從而減少工作負載的影響範圍。如需詳細資訊，請參閱有關此方法的 AWS re：Invent 簡報。

您可以結合碎片方法與主要待命方法，為碎片提供容錯移轉功能。您需要將經過測試的容錯移轉程序設計成工作負載，以及資料對帳的程序，以確保容錯移轉後資料存放區的交易一致性。本指南稍後會詳細說明這些內容。

關鍵指引

- 當發生故障時，等待複寫的寫入不會遞交至待命區域的可能性很高。在複寫恢復之前，資料將無法使用（假設為非同步複寫）。
- 在容錯移轉過程中，將需要資料對帳程序，以確保使用非同步複寫的資料存放區維持交易一致狀態。這需要特定的商業邏輯，而不是由資料存放區本身處理的邏輯。
- 需要強式一致性時，需要修改工作負載，以容忍同步複寫的資料存放區所需的延遲。

多區域基礎 3：了解工作負載相依性

特定工作負載在區域中可能有數個相依性，例如 AWS 服務 已使用、內部相依性、第三方相依性、網路相依性、憑證、金鑰、秘密和參數。為了確保在故障案例期間操作工作負載，主要區域和待命區域之間應該沒有相依性；每個區域都應該能夠獨立運作。若要達成此目的，請仔細檢查工作負載中的所有相依性，以確保它們在每個區域中可用。這是必要的，因為主要區域中的失敗不應影響待命區域。此外，您必須了解當相依性處於降級狀態或完全無法使用時，工作負載的運作方式，以便您可以設計適當的解決方案來處理此問題。

3.a： AWS 服務

當您設計多區域架構時，請務必了解將使用 AWS 服務 的 、這些服務的多區域功能，以及您需要設計哪些解決方案才能實現多區域目標。例如，Amazon Aurora 和 Amazon DynamoDB 可以將資料非同步複寫到待命區域。工作負載將從中執行的所有區域都需要提供所有 AWS 服務 相依性。若要確認您使用的服務可在所需區域中使用，請檢閱AWS 服務 依區域清單。

3.b：內部和第三方相依性

確保每個工作負載的內部相依性在其操作所在的區域中可用。例如，如果工作負載由許多微服務組成，請識別構成商業功能的所有微服務，並確認所有這些微服務都部署在工作負載運作所在的每個區域中。或者，定義策略來正常處理無法使用的微服務。

不建議在工作負載內的微服務之間進行跨區域呼叫，並應維持區域隔離。這是因為建立跨區域相依性會增加相互關聯失敗的風險，這會抵銷工作負載隔離區域實作的優勢。內部部署相依性也可能是工作負載的一部分，因此請務必了解如果主要區域發生變更，這些整合的特性可能會如何變更。例如，如果待命區域離現場部署環境較遠，增加的延遲可能會產生負面影響。

了解軟體即服務 (SaaS) 解決方案、軟體開發套件 (SDKs) 和其他第三方產品相依性，並能夠練習這些相依性降級或無法使用的情況，將有助於更深入地了解系統鏈在不同故障模式下的運作和行為。這些相依性可能在您的應用程式程式碼內，例如使用 在外部管理秘密AWS Secrets Manager，或者可能涉及第三方保存庫解決方案 （ 例如 HashiCorp)，或依賴 進行聯合登入AWS IAM Identity Center的身分驗證系統。

在相依性方面具有備援可以提高彈性。如果 SaaS 解決方案或第三方相依性使用 AWS 區域 與工作負載相同的主要節點，請與廠商合作，判斷其彈性狀態是否符合您的工作負載需求。

此外，請注意工作負載與其相依性之間的共同命運，例如第三方應用程式。如果在容錯移轉後次要區域中 （ 或從中 ） 無法使用相依性，工作負載可能無法完全復原。

3.c：容錯移轉機制

DNS 通常用作容錯移轉機制，將流量從主要區域轉移到待命區域。嚴格檢閱和仔細檢查容錯移轉機制所需的所有相依性。例如，如果您的工作負載使用 Amazon Route 53，了解控制平面託管於 ，us-east-1表示您正在對該特定區域中的控制平面取得相依性。如果主要區域也是us-east-1因為它建立單一故障點，則不建議將其作為容錯移轉機制的一部分。如果您使用另一個容錯移轉機制，您應該深入了解容錯移轉無法如預期運作的情況，然後視需要規劃應變或開發新的機制。檢閱部落格文章使用 Amazon Route 53 建立災難復原機制，以了解您可以用來成功容錯移轉的方法。

如上一節所述，屬於業務功能的所有微服務都需要在部署工作負載的每個區域中提供。作為容錯移轉策略的一部分，所有屬於業務能力的微服務都應一起容錯移轉，以消除跨區域呼叫的機會。或者，如果微服務獨立容錯移轉，則可能會有不良行為，例如微服務可能會進行跨區域呼叫。這會導致延遲，並可能導致工作負載在用戶端逾時期間變得無法使用。

3.d：組態相依性

憑證、金鑰、秘密、Amazon Machine Image (AMIs)、容器映像和參數是設計多區域架構時所需的相依性分析的一部分。在可能的情況下，最好將這些元件當地語系化，以便它們在這些相依性的區域之間沒有共同命運。例如，您應該變更憑證的過期日期，以防止過期憑證（警示設定為「預先通知」）影響多個區域的案例。

加密金鑰和秘密也應該是區域特定的。如此一來，如果金鑰或秘密的輪換發生錯誤，則影響僅限於特定區域。

最後，任何工作負載參數都應儲存在本機，以便在特定區域中擷取工作負載。

關鍵指引

- 多區域架構受益於區域之間的實體和邏輯分離。應用程式層的跨區域相依性簡介會破壞此優勢。避免此類相依性。
- 容錯移轉控制應該在主要區域上沒有相依性的情況下運作。
- 容錯移轉應跨使用者旅程進行協調，以消除跨區域呼叫延遲增加和相依性增加的可能性。

多區域基本 4：營運準備度

操作多區域工作負載是一項複雜的任務，伴隨多區域架構特有的操作挑戰。這些包括 AWS 帳戶 管理、重組部署程序、建立多區域可觀測性策略、建立和測試復原程序，然後管理成本。營運準備度審查 (ORR) 可協助團隊準備用於生產的工作負載，無論是在單一區域還是跨多個區域執行。

4.a： AWS 帳戶 管理

若要跨 部署工作負載 AWS 區域，請確保跨區域帳戶內所有AWS 服務 配額的同位。首先，識別屬於架構的所有 AWS 服務 ，查看待命區域中的計劃用量，然後將計劃用量與目前的用量進行比較。在某些情況下，如果之前未使用待命區域，您可以參考預設的服務配額來了解起點。然後，在將使用的所有服務中，使用 Service Quotas 主控台 （需要登入） 或 APIs 請求提高配額。

在每個區域中設定 AWS Identity and Access Management (IAM) 角色，以提供操作員、自動化工具和待命區域內資源 AWS 服務 的適當許可。若要實現多區域架構的區域隔離，請依區域隔離角色。在使用待命區域上線之前，請確定有適當的許可。

4.b：部署實務

多區域功能可能會使將工作負載部署到多個區域變得複雜。您需要確保一次部署到一個區域。例如，如果您使用主動-被動的方法，您應該先部署到主要區域，然後再部署到待命區域。 AWS CloudFormation可協助您將基礎設施部署到單一或多個區域，並可根據您的需求量身打造。 AWS CodePipeline可協助您建置持續整合/持續交付 (CI/CD) 管道，其具有跨區域動作，允許部署到與管道所在區域不同的區域。這結合了藍/綠等強大的部署策略，允許最低到零的停機時間部署。

不過，當應用程式或資料的狀態未外部化至持久性存放區時，狀態功能的部署可能會變得更加複雜。在這些情況下，請仔細量身打造部署程序以符合您的需求。設計部署管道和程序，以一次部署到一個區域，而不是同時部署到多個區域。這可減少區域之間相互關聯失敗的機率。若要了解 Amazon 用來自動化軟體部署的技巧，請參閱 AWS 建置者程式庫文章自動化安全的實作部署。

4.c：可觀測性

當您設計多區域時，請考慮如何監控每個區域中所有元件的運作狀態，以取得區域運作狀態的整體檢視。這可能包括監控複寫延遲的指標，這不是單一區域工作負載的考量。

當您建置多區域架構時，請考慮也從待命區域觀察工作負載的效能。這包括從待命區域執行運作狀態檢查和 Canary （合成測試），以提供主要區域運作狀態的外部檢視。此外，您可以使用 Amazon

CloudWatch 網路監視器，從最終使用者的角度了解外部網路的狀態和工作負載的效能。主要區域應具有相同的可觀測性，以監控待命區域。

來自待命區域的 Canary 應監控客戶體驗指標，以判斷工作負載的整體運作狀態。這是必要的，因為如果主要區域發生問題，主要區域中的可觀測性可能會受損，並會影響您評估工作負載運作狀態的能力。

在這種情況下，觀察該區域以外的 可以提供洞見。這些指標應彙總到每個區域中可用的儀表板，以及每個區域中建立的警示。由於 CloudWatch 是區域性服務，因此在兩個區域中都具有警示是必要的。此監控資料將用於讓呼叫從主要區域容錯移轉到待命區域。

4.d：程序和程序

回答問題的最佳時機：「我應該何時容錯移轉？」 在您需要之前很長。在問題發生之前預先定義包含人員、程序和技術的復原計畫，並定期進行測試。決定復原決策架構。如果有精準的復原程序，而且充分了解復原的時間，您可以選擇使用符合 RTO 目標的容錯移轉來啟動復原程序。識別主要區域中的應用程式問題之後，可能會立即發生此時間點，或者當區域中應用程式內的復原選項已用盡時，可能會進一步進入事件。

容錯移轉動作本身應該 100% 自動化，但啟動容錯移轉的決定應由人類做出，通常是組織中少數的預定人員。這些人員應考慮資料遺失和事件的相關資訊。此外，需要明確定義容錯移轉的條件，並在組織內全面了解。若要定義和完成這些程序，您可以使用 AWS Systems Manager Runbook，以允許完整的end-to-end自動化，並確保在測試和容錯移轉期間執行的程序一致性。

這些 Runbook 應該可在主要和待命區域中使用，以啟動容錯移轉或容錯回復程序。當此自動化準備就緒時，請定義並遵循定期測試節奏。這可確保發生實際事件時，回應會遵循組織可信的明確定義、實務程序。考慮資料對帳程序的既定公差也很重要。確認提議的程序符合已建立的 RPO/RTO 要求。

4.e：測試

擁有未經測試的復原方法等於沒有復原方法。基本的測試層級是執行復原程序，以切換應用程式的操作區域。有時這稱為應用程式輪換方法。我們建議您建置將區域切換為正常操作狀態的功能；不過，僅此測試還不夠。

彈性測試對於驗證應用程式的復原方法也很重要。這包括注入特定失敗案例、了解您的應用程式和復原程序如何反應，然後在測試未按計劃進行時實作所需的任何緩解措施。在沒有錯誤的情況下測試復原程序，並不會告訴您應用程式在發生錯誤時的整體行為。您必須制定計劃，以根據預期的失敗案例來測試復原。 AWS Fault Injection Service提供越來越多的案例清單，協助您開始使用。

這對於高可用性應用程式尤其重要，因為需要嚴格測試以確保符合業務連續性目標。主動測試復原功能可降低生產失敗的風險，進而建立應用程式可達到所需限制復原時間的信心。定期測試也會建立營運專

業知識，這可讓團隊在中斷發生時快速可靠地從中斷中復原。行使復原方法的人工元素或程序與技術層面同樣重要。

4.f：成本和複雜性

多區域架構的成本影響取決於更高的基礎設施用量、營運開銷和資源時間。如前所述，待命區域中的基礎設施成本與預先佈建時主要區域中的基礎設施成本類似，因此會加倍您的總成本。佈建容量，使其足以進行日常操作，但仍保留足夠的緩衝容量來容忍需求激增。然後，在每個區域中設定相同的限制。

此外，如果您採用主動-主動架構，您可能需要進行應用程式層級的變更，才能在多區域架構中成功執行應用程式。這些變更可能需要耗費大量時間和資源，才能設計和操作。組織至少需要花時間了解每個區域的技術和業務相依性，以及設計容錯移轉和容錯回復程序。

團隊也應該進行正常的容錯移轉和容錯回復練習，以熟悉將在事件期間使用的 Runbook。雖然這些練習對於從多區域投資取得預期成果至關重要，但它們代表機會成本，並從其他活動佔用時間和資源。

4.g：組織多區域容錯移轉策略

AWS 區域 提供故障隔離界限，以防止相關故障，並在 AWS 服務 故障發生時包含對單一區域造成的影響。您可以使用這些錯誤界限來建置多區域應用程式，這些應用程式由每個區域中獨立的錯誤隔離複本組成，以限制共用命運案例。這可讓您建置多區域應用程式，並使用各種方法，從備份和還原到指示燈，再到主動-主動，以實作多區域架構。不過，應用程式通常不會單獨運作，因此請將您將使用的元件及其相依性視為容錯移轉策略的一部分。一般而言，多個應用程式會共同支援使用者案例，這是提供給最終使用者的特定功能，例如在社交媒體應用程式上張貼圖片和字幕，或在電子商務網站上查看。因此，您應該制定組織多區域容錯移轉策略，以提供必要的協調性和一致性，讓您的方法成功。

組織可以從中挑選四種高階策略，以引導多區域方法。這些是從最精細到最廣泛的方法列出：

- 元件層級容錯移轉
- 個別應用程式容錯移轉
- 相依性圖表容錯移轉
- 整個應用程式產品組合容錯移轉

每個策略都有權衡並解決不同的挑戰，包括容錯移轉決策的靈活性、測試容錯移轉組合的能力、模式行為的存在，以及組織在規劃和實作方面的投資。若要深入了解每個策略，請參閱 AWS 部落格文章建立組織多區域容錯移轉策略。

關鍵指引

- 檢閱所有 AWS 服務 配額，以確保它們在工作負載將運作的所有區域中是相同的。
- 部署程序應該一次以一個區域為目標，而不是同時涉及多個區域。
- 複寫延遲等其他指標專屬於多區域案例，應加以監控。
- 將工作負載的監控延伸到主要區域之外。監控每個區域的客戶體驗指標，並從執行工作負載的每個區域外部測量此資料。
- 定期測試容錯移轉和容錯回復。實作容錯移轉和容錯回復程序的單一 Runbook，並將其用於測試和即時事件。用於測試和即時事件的 Runbook 不應不同。
- 了解容錯移轉策略的權衡。實作相依性圖表或整個應用程式產品組合策略。

結論和資源

本指南涵蓋多區域架構的常見使用案例、實作這些架構的基礎知識，以及此方法的影響。您可以將這些基礎知識套用至任何工作負載，並使用資訊做為架構，以協助判斷多區域架構是否適合您的業務。

如需詳細資訊，請參閱下列資源：

- AWS 架構中心
- AWS Well-Architected 架構
- AWS Well-Architected Tool
- 建立組織多區域容錯移轉策略 (AWS 部落格文章）
- AWS 多區域功能 (AWS re：Post 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	整份指南的更新。	2024 年 12 月 27 日
初次出版	—	2022 年 12 月 20 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

在整個生命週期中追蹤資料的原始伺服器和歷史記錄的程序，例如資料的產生、傳輸和儲存方式。

資料主體

正在收集和處理其資料的個人。

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並排定優先順序。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)

組織之間商業文件的自動交換。如需詳細資訊，請參閱什麼是電子資料交換。

加密

一種運算程序，可將人類可讀取的純文字資料轉換為加密文字。

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 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

一種使用頻繁和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。對於需要特定格式、推理或網域知識的任務，少量的提示可以有效。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，已針對廣義和未標記資料的大量資料集進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及自然語言的交談。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)

當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。

可程式設計邏輯控制器 (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

請參閱擷取增強產生。

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 個 R。

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)

服務中斷與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。

重新定位

請參閱 7 個 R。

Replatform

請參閱 7 個 R。

回購

請參閱 7 個 R。

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

矩陣，定義所有參與遷移活動和雲端操作之各方的角色和責任。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 個 R。

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。此功能會啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS Management Console 或呼叫 AWS API 操作，而不必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？。

依設計的安全性

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)

服務效能方面的測量，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。

單點故障 (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，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。

暖資料

不常存取的資料。查詢這類資料時，通常可接受中等緩慢的查詢。

視窗函數

SQL 函數，對與目前記錄在某種程度上相關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，並防止資料遭到刪除或修改。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瑕疵或漏洞。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。