為多租戶軟體 SaaS 應用程式實作受管理的 PostgreSQL AWS

AWS 規定指引

AWS 規定指引: 為多租戶軟體 SaaS 應用程式實作受管理的 PostgreSQL AWS

Table of Contents

為多租戶軟體 SaaS 應用程式實作受管理的 PostgreSQL AWS

泰比沃德和托馬斯·戴維斯，Amazon Web Services（ ）AWS

2024 年 4 月 (文件歷史記錄)

當您選取要儲存作業資料的資料庫時，請務必考慮資料的結構化方式、資料將回應的查詢、提供答案的速度，以及資料平台本身的彈性。除了這些一般考量之外，軟體即服務 (SaaS) 對作業資料的影響，例如效能隔離、租用戶安全性，以及多租戶 SaaS 應用程式典型資料的獨特特性和設計模式。本指南討論這些因素如何適用於使用 Amazon Web Services (AWS) 上的 PostgreSQL 資料庫作為多租戶 SaaS 應用程式的主要操作資料存放區。具體而言，本指南著重於兩個 AWS 受管 PostgreSQL 選項：Amazon Aurora PostgreSQL 相容版本和適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS)。

目標業務成果

本指南針對使用與 Aurora PostgreSQL 相容的多租戶 SaaS 應用程式提供最佳實務的詳細分析。我們建議您使用本指南中提供的設計模式和概念，為多租戶 SaaS 應用程式提供資訊和標準化您的 Aurora PostgreSQL 相容或 Amazon RDS (適用於 PostgreSQL) 的實作。

此規範指引有助於達成下列業務成果：

- 為您的使用案例選擇最佳的 AWS 受管 PostgreSQL 選項 — 本指南將資料庫使用情況的關聯式和非關聯式選項與 SaaS 應用程式進行比較。它還討論了哪些使用案例最適用於 Aurora 與 PostgreSQL 相容和 Amazon RDS for PostgreSQL。此資訊將有助於為您的 SaaS 應用程式選擇最佳選項。
- 透過採用 SaaS 分割模型來強制實施 SaaS 最佳實務 — 本指南討論並比較適用於 PostgreSQL 資料庫管理系統 (DBMS) 的三種廣泛 SaaS 分割模型：集區、橋接和筒倉模型及其變化。這些方法可擷取 SaaS 最佳實務，並在設計 SaaS 應用程式時提供彈性。實施 SaaS 分區模型是保留最佳實踐的關鍵部分。
- 在集區 SaaS 磁碟分割模型中有效使用 RLS — 資料列層級安全性 (RLS) 透過限制可根據使用者或內容變數檢視的資料列，來支援在單一 PostgreSQL 表格中強制執行租用戶資料隔離。當您使用集區磁碟分割模式時，需要 RLS 才能防止跨租用戶存取。

選取 SaaS 應用程式的資料庫

對於許多多租戶 SaaS 應用程式而言，選取操作資料庫可以精簡為關聯式和非關聯式資料庫之間的選擇，或是兩者的組合。若要做出決定，請考慮下列高階應用程式資料需求和特性：

- 應用程式的資料模型
- 資料的存取模式
- 資料庫延遲需求
- 數據完整性和事務完整性要求（原子性，一致性，隔離性和耐久性或 ACID）
- 跨區域可用性與回復需求

下表列出應用程式資料需求和特性，並在資料 AWS 庫產品的內容中討論這些需求：與 Aurora PostgreSQL 相容和 Amazon RDS (關聯式) 和 Amazon DynamoDB (非關聯式)。當您嘗試在關聯式和非關聯式作業資料庫產品之間做決定時，您可以參考此矩陣。

資料庫	SaaS 應用程式資料需求與特性				
	資料模型	存取模式	延遲需求	資料和交易完整性	跨區域可用性與回復
关系 （Aurora 兼容和亞馬遜 RDS 適用於 PostgreSQL）	關係或高度規範化。	不必事先進行徹底的計劃。	最好是更高的延遲容忍度；在預設情況下，使用 Aurora 並實作僅供讀取複本、快取和類似功能，可達到較低的延遲。	默認情況下維護高數據和交易完整性。	在 Amazon RDS 中，您可以建立跨區域擴展和容錯移轉的僅供讀取複本。Aurora 主要是自動化這個過程。對於跨多個主動-主動組態 AWS 區域，您可以將寫轉送與 Aurora

					全域資料庫搭配使用。
非关系 (Amazon DynamoDB)	通常非標準化。這些資料庫利用模式來建many-to-many立關係、大型項目和時間序列資料的模式。	在產生資料模型之前，必須徹底瞭解資料的所有存取模式 (查詢)。	具有 Amazon DynamoDB 加速器 (DAX) 等選項的極低延遲，可進一步提升效能。	以效能為代價的選擇性交易完整性。數據完整性考慮轉移到應用程序。	透過全域表，輕鬆進行跨區域復原和主動-主動組態。（ACID 合規性只能在單個 AWS 區域中實現。）

某些多租用戶 SaaS 應用程式可能具有唯一的資料模型或特殊情況，這些特殊情況會由前一個資料表中未包含的資料庫提供更 例如，時間序列資料集、高度連線的資料集或維護集中式交易分類帳可能需要使用不同類型的資料庫。分析所有可能性已超出本指南的範圍。有關 AWS 資料庫產品的完整清單，以及它們如何能夠在高層級滿足不同的使用案例，請參閱 Amazon Web Services 概觀白皮書中的資料庫一節。

本指南的其餘部分著重於支援 PostgreSQL 的 AWS 關聯式資料庫服務：與 Amazon RDS 和 Aurora PostgreSQL 相容。DynamoDB 需要不同的方法來針對 SaaS 應用程式進行最佳化，這超出本指南的範圍。如需有關 DynamoDB 的詳細資訊，請參閱 AWS 部落格文章使用 Amazon DynamoDB 分割集區的多租戶 SaaS 資料。

Amazon RDS 和 Aurora 之間的選擇

在大多數情況下，我們建議您使用相容於亞馬遜 RDS 的 Aurora PostgreSQL。下表顯示在這兩個選項之間做出決定時應考慮的因素。

数据库管理系统	Amazon RDS for PostgreSQL	Aurora 郵政兼容
擴充性	複寫延遲時間為分鐘，最多 5 個僅供讀取複本	複寫延遲不到一分鐘 (通常在全域資料庫少於 1 秒)，最多 15 個僅供讀取複本
崩潰恢復	檢查點相隔 5 分鐘（默認情況下）可能會降低數據庫性能	以 parallel 執行緒進行非同步復原，可快速

数据库管理系统	Amazon RDS for PostgreSQL	Aurora 郵政兼容
故障轉	60-120 秒，除了崩潰恢復時間	通常約 30 秒（包括崩潰恢復）
儲存	最多 25 萬六千個 IOPS	IOPS 僅受到 Aurora 執行個體大小和容量的限制
高可用性和災難復原	兩個具備待命執行個體的可用區域、跨區域容錯移轉至僅供讀取複本或複製備份	預設有三個可用區域、跨區域容錯移轉與 Aurora 全域資料庫、主動-主動組態的AWS 區域 寫入轉送
備份	在備份時段期間，可能會影響效能	自動增量備份，不影響效能
資料庫實例類別	查看 Amazon RDS 執行個體類別清單	查看 Aurora 執行個體類別清單

在上表中描述的所有類別中，Aurora PostgreSQL 兼容通常是更好的選擇。不過，Amazon RDS for PostgreSQL 對於中小型工作負載來說仍然有意義，因為它具有更多的執行個體類別選擇，可能會提供更符合成本效益的選項，但會犧牲 Aurora 更強大的功能集。

適用於 PostgreSQL 的多租戶 SaaS 分割模式

完成多租戶的最佳方法取決於 SaaS 應用程式的需求。以下各節示範了在 PostgreSQL 中成功實作多租戶的分割模型。

> **Note**
>
> 本節中討論的模型適用於 Amazon RDS for PostgreSQL 版和 Aurora 兼容。本節中對 PostgreSQL 的參照適用於這兩種服務。

您可以在 PostgreSQL 中使用三種高階模型進行 SaaS 磁碟分割：筒倉、橋接器和集區。下圖概述了筒倉和池模型之間的權衡。橋接模型是筒倉和泳池模型的混合體。

分割模型	優點	缺點
筒倉	• 合規 • 沒有跨租戶影響 • 承租人層級調整 • 承租人層級的可用性	• 受到敏捷 • 無集中式管理 • 部署複雜度 • 費用
游泳池	• 敏捷 • 成本最佳化 • 集中式管理 • 簡化部署	• 跨租用戶影響 • 合規性挑戰 • 全部或全部可用性
橋接器	• 一些合規性 • 敏捷 • 成本最佳化 • 集中式管理	• 一些合規性挑戰 • 全部或全無可用性（主要） • 跨租用戶影響 • 部署複雜度

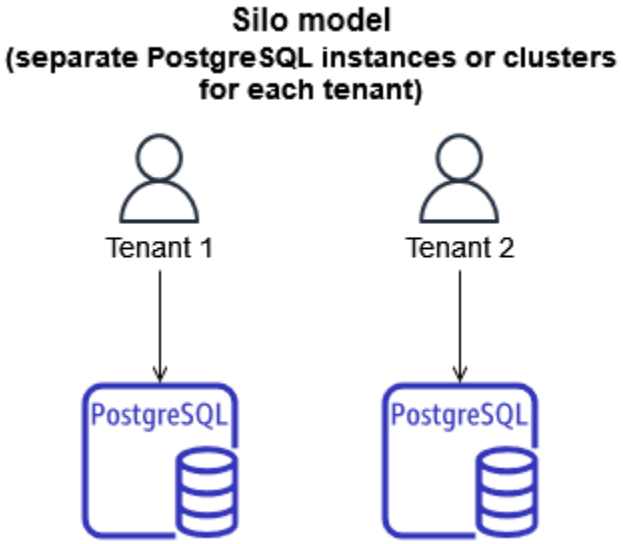
以下章節將更詳細地討論每個模型。

分割模型：

- PostgreSQL 筒倉模型

PostgreSQL 筒倉模型

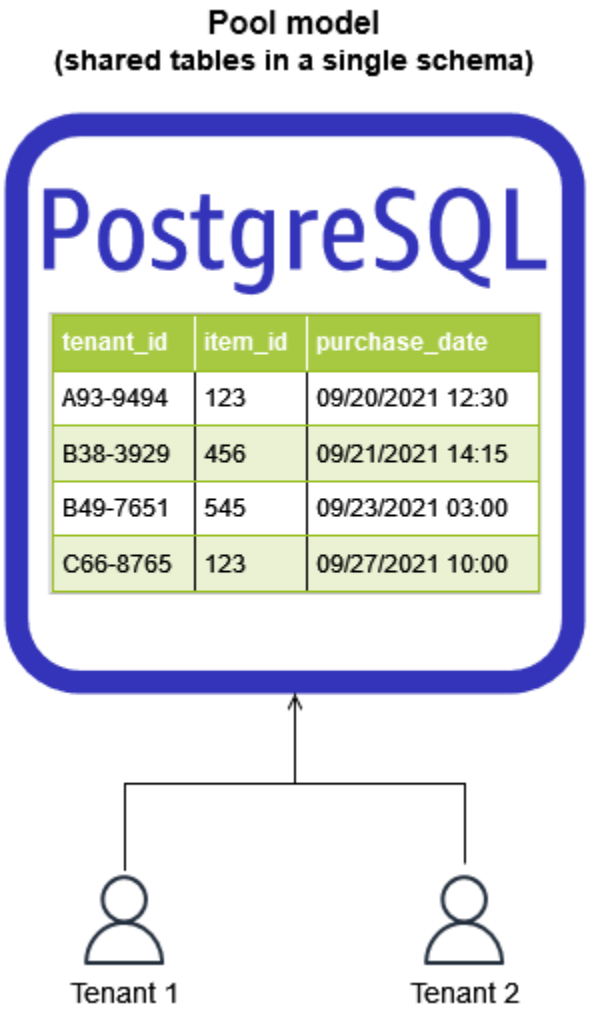
透過為應用程式中的每個租用戶佈建 PostgreSQL 執行個體來實作筒倉模型。筒倉模型在租戶性能和安全隔離方面表現出色，並完全消除了嘈雜的鄰居現象。當一個租用戶對系統的使用影響另一個租用戶的性能時，就會發生嘈雜的鄰居現象。筒倉模型可讓您針對每個租用戶量身打造效能，並可能將中斷限制為特定租用戶的筒倉。但是，通常驅動採用筒倉模型的原因是嚴格的安全性和法規限制。這些限制可以由 SaaS 客戶推動。例如，SaaS 客戶可能會因內部限制而要求將其資料隔離，而 SaaS 供應商可能需要額外付費提供此類服務。



儘管在某些情況下可能需要倉庫模型，但它有許多缺點。通常很難以符合成本效益的方式使用筒倉模型，因為管理多個 PostgreSQL 執行個體之間的資源消耗可能很複雜。此外，此模型中資料庫工作負載的分散式本質使得維護租用戶活動的集中檢視變得更加困難。管理這麼多獨立操作的工作負載會增加營運和管理開銷。由於您必須佈建承租人特定的資源，因此筒倉模型也會使租用戶上線變得更加複雜且耗時。此外，整個 SaaS 系統可能難以擴展，因為特定於承租人的 PostgreSQL 執行個體數量不斷增加，將需要更多的操作時間來管理。最後一個考慮因素是，應用程式或資料存取層必須維護租用戶與其關聯 PostgreSQL 執行個體的對應，這會增加實作此模型的複雜性。

PostgreSQL 集區模型

集區模型的實作方式是佈建單一 PostgreSQL 執行個體 (Amazon RDS 或 Aurora)，並使用列層級安全性 (RLS) 來維護租用戶資料隔離。RLS 原則會限制SELECT查詢傳回資料表中的哪些資料列，或哪些資料列受INSERTUPDATE、和DELETE命令影響。集區模型將所有租用戶資料集中在單一 PostgreSQL 結構描述中，因此更具成本效益，並且需要較少的維護作業負荷。由於其集中化，監控此解決方案也明顯簡單。但是，監視集區模型中的承租人特定影響通常需要在應用程式中進行一些額外的儀器。這是因為 PostgreSQL 預設不知道哪個租用戶正在消耗資源。由於不需要新的基礎結構，所以可以簡化租用戶上線。這種敏捷性可讓您更輕鬆地完成快速且自動化的租用戶上線工作流程。



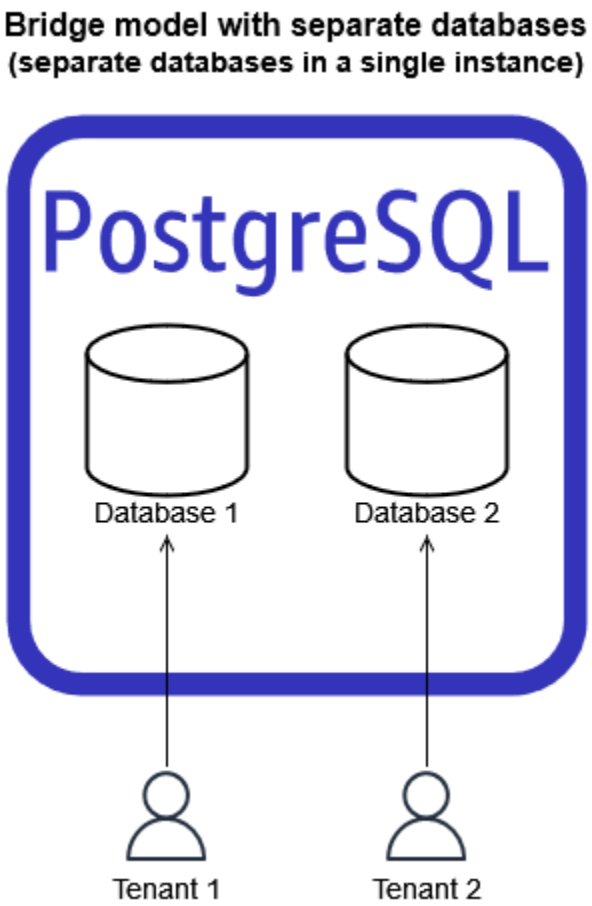
雖然池模型通常更具成本效益並且管理更簡單，但它確實有一些缺點。在泳池模型中無法完全消除嘈雜的鄰居現象。不過，可以透過確保 PostgreSQL 執行個體上有適當的資源，以及使用策略減少 PostgreSQL 中的負載 (例如將查詢卸載到僅供讀取複本或 Amazon) 來減輕此問題 ElastiCache。有效監視在回應租用戶效能隔離問題方面也扮演著重要角色，因為應用程式檢測可以記錄和監視承租人特定

的活動。最後，某些 SaaS 客戶可能不會發現 RLS 提供的邏輯分離足夠，並且可能會要求採取額外的隔離措施。

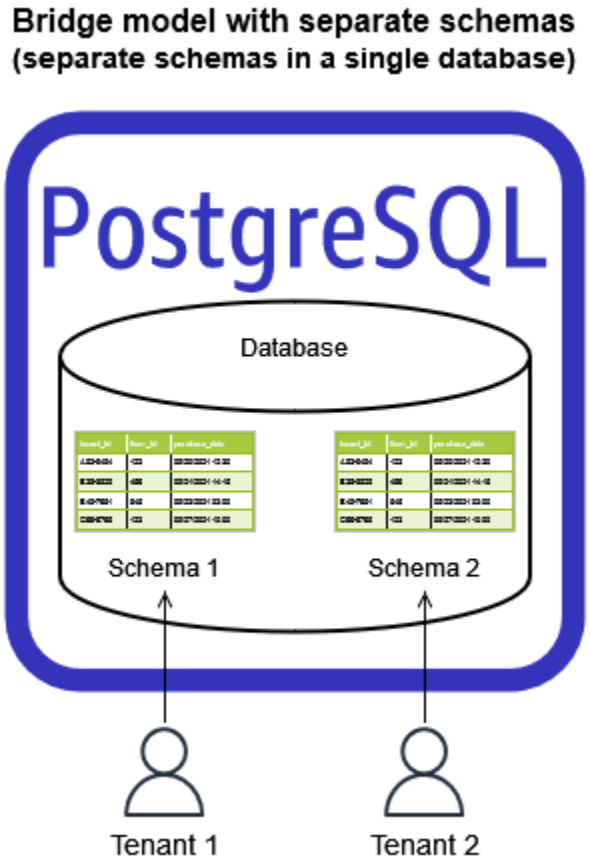
PostgreSQL 橋接模型

PostgreSQL 橋接器模型是集區和孤立方法的組合。就像集區模型一樣，您可以為每個租用戶佈建單一 PostgreSQL 執行個體。若要維護租用戶資料隔離，您可以使用 PostgreSQL 邏輯結構。在下圖中，PostgreSQL 數據庫用於邏輯上分離數據。

> **Note**
>
> PostgreSQL 資料庫不會指向個別的 Amazon RDS for PostgreSQL) 或相容於 Aurora 的資料庫執行個體。相反，它是指 PostgreSQL 數據庫管理系統的邏輯構造來分隔數據。



您也可以使用單一 PostgreSQL 資料庫來實作橋接器模型，並在每個資料庫中包含承租人特定結構描述，如下圖所示。



橋接器模型遭受與集區模型相同的嘈雜鄰居和租用戶效能隔離問題。它還需要針對每個租用戶進行佈建個別的資料庫或結構描述，因此會產生一些額外的作業和佈建額外負荷。它需要有效的監控才能快速回應租用戶效能問題。它也需要應用程式儀器來監視承租人特定的使用情況。整體而言，橋接器模型可視為 RLS 的替代方案，透過需要新的 PostgreSQL 資料庫或結構描述，稍微增加租用戶上線工作量。與筒倉模型一樣，應用程式或資料存取層必須維護租用戶與其關聯 PostgreSQL 資料庫或結構描述的對應。

決策矩陣

若要決定您應該搭配 PostgreSQL 使用哪種多租用戶 SaaS 分割模型，請參閱下列決策矩陣。矩陣會分析下列四個分割選項：

- 筒倉 — 每個租用戶的個別 PostgreSQL 執行個體或叢集。
- 具有不同資料庫的橋接 — 單一 PostgreSQL 執行個體或叢集中的每個租用戶都有個別的資料庫。

- 具有不同結構描述的橋接 — 單一 PostgreSQL 資料庫、單一 PostgreSQL 執行個體或叢集中的每個租用戶的個別結構描述。
- 集區 — 單一執行個體和結構描述中租用戶的共用資料表。

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
使用案例	完全控制資源使用情況的資料隔離是一項關鍵要求，或者您擁有非常龐大且效能非常敏感的租用戶。	數據隔離是一項關鍵要求，租戶的數據是有限或不需要交叉引用。	具有中等數據量的中等租戶數量。如果您必須交叉引用租戶的數據，這是首選的模式。	大量租戶，每個租戶的數據較少。
新的租戶上線敏捷性	非常慢。每個租用戶都需要新的執行個體或叢集。)	適中慢。(需要為每個租用戶建立新資料庫以儲存結構描述物件。)	適中慢。(需要為每個承租人建立新結構描述以儲存物件。)	最快的選擇。(需要最低限度的設定。)
資料庫連線集區組態工作與效率	需要大量的努力。(每個租用戶有一個連線集區。) 效率較低。（租用戶之間沒有數據庫連接共享。）	需要大量的努力。(每個租用戶有一個連線集區組態，除非您使用 Amazon RDS 代理伺服器。) 效率較低。(租用戶與連線總數之間沒有資料庫連線共用。 所有租用戶的使用量是基於資料庫執行個體類別的限制。)	所需的努力更少。(適用於所有租用戶的一個連線集區組態。) 效率適中。(僅在工作階段集區模式下透過SET ROLE或SET SCHEMA命令重複使用連線。 SET使用 Amazon RDS Proxy 時，命令也會導致工作階段釘選，但可以消除用戶端連線	最少的努力要求。 最有效。(一個連線集區供所有租用戶使用，並有效率地重複使用所有租用戶。 資料庫連線限制是基於資料庫執行個體類別。)

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
			集區，並為每個請求建立直接連線以提高效率。）	
數據庫維護（真空管理）和資源使用	更簡單的管理。	中度複雜。（可能會導致高資源消耗，因為之後必須為每個數據庫啟動真空工作者`vacuum_naptime` ，從而導致高自動真空啟動器 CPU 使用率。 對於每個數據庫的 PostgreSQL 系統目錄表進行吸塵，也可能會產生額外的開銷。）	大型 PostgreSQL 系統目錄資料表。（總`pg_catalog` 大小以數十 GB 為單位，具體取決於租戶的數量和關係。 可能需要對真空相關參數進行修改，以控制表膨脹。）	資料表可能很大，具體取決於每個租用戶的租用戶數量和資料。（可能需要對真空相關參數進行修改，以控制表膨脹。）
擴充功能的管理	顯著的努力（在單獨的實例中的每個數據庫）。	顯著的努力（在每個數據庫級別）。	最小的努力（在公共數據庫中一次）。	最小的努力（在公共數據庫中一次）。
變更部署工作量	重大努力。（Connect 到每個單獨的實例並推出更改。）	重大努力。（Connect 到每個數據庫和模式，並推出更改。）	適度的努力。（Connect 到通用數據庫並為每個模式發布更改。）	最小的努力。（Connect 到通用數據庫並推出更改。）
變更部署 — 影響範圍	極小。（單一租戶受影響。）	極小。（單一租戶受影響。）	極小。（單一租戶受影響。）	非常大。（受影響的所有租戶。）

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
查詢績效管理和工作量	可管理的查詢效能。	可管理的查詢效能。	可管理的查詢效能。	維護查詢效能可能需要大量的努力。(隨著時間的推移，查詢執行速度可能會因為資料表的大小增加而變慢。 您可以使用表分區和數據庫分片來保持性能。)
跨租用戶資源影響	沒有影響。（租戶之間沒有資源共享。)	適中 (租用戶會共用一般資源，例如執行個體 CPU 和記憶體)。	適中 (租用戶會共用一般資源，例如執行個體 CPU 和記憶體)。	重大影響。(租用戶在資源、鎖定衝突等方面相互影響。)
承租人層級調整 (例如，針對特定承租人建立其他索引，或針對特定承租人建立資料庫參數調整)	可能。	有可能。可以為每個承租人進行結構描述層級變更，但資料庫參數在所有承租人之間都是全域的。)	有可能。可以為每個承租人進行結構描述層級變更，但資料庫參數在所有承租人之間都是全域的。)	不可能。（表由所有租戶共享。)

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
重新平衡效能敏感租用戶的工作量	極小。（無需重新平衡。 擴充伺服器和 I/O 資源以處理此案例。）	適中。(使用邏輯複寫或pg_dump匯出資料庫，但停機時間可能會因資料大小而定。 您可以使用 Amazon RDS for PostgreSQL 中的可傳輸資料庫功能，更快速地在執行個體之間複製資料庫。)	中度但可能涉及冗長的停機時間。(使用邏輯複寫或pg_dump匯出結構描述，但停機時間可能會因資料大小而定。)	重要的是，因為所有租戶共享相同的表。（分片數據庫需要將所有內容複製到另一個實例，以及清理租戶數據的附加步驟。 ） 最有可能需要改變應用程序邏輯。
主要版本升級的資料庫停機	標準停機 (取決於 PostgreSQL 系統目錄大小。)	停機時間可能更長。（根據系統目錄大小，時間會有所不同。 PostgreSQL 系統目錄表格也會跨資料庫複製)	停機時間可能更長。(視 PostgreSQL 系統目錄大小而定，時間會有所不同。)	標準停機 (取決於 PostgreSQL 系統目錄大小。)
管理額外負荷 (例如，用於資料庫記錄分析或備份工作監督)	重大投入	最小的努力。	最小的努力。	最小的努力。
承租人層級的可用性	最高（每個租戶失敗並獨立恢復。 ）	更高的影響範圍。(如果發生硬體或資源問題，所有租用戶都會失敗並一起復原。)	更高的影響範圍。(如果發生硬體或資源問題，所有租用戶都會失敗並一起復原。)	更高的影響範圍。(如果發生硬體或資源問題，所有租用戶都會失敗並一起復原。)

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
承租人層級備份與回復工作	最低有效。(每個租用戶可以獨立備份和還原。)	適度的努力。(針對每個承租人使用邏輯匯出和匯入。 一些編碼和自動化是必需的。)	適度的努力。(針對每個承租人使用邏輯匯出和匯入。 一些編碼和自動化是必需的。)	重大努力。（所有租戶共享相同的表格。)
租戶級別的 point-in-time 恢復工作	最小的努力。(使用快照使用點入時間復原，或在 Amazon Aurora 中使用回溯功能。)	適度的努力。（使用快照恢復，然後導出/導入。 但是，這將是一個緩慢的操作。)	適度的努力。（使用快照恢復，然後導出/導入。 但是，這將是一個緩慢的操作。)	巨大的努力和複雜性。
統一結構描述	每個租用戶的結構描述名稱相同。	每個租用戶的結構描述名稱相同。	每個租用戶的不同結構描述。	通用結構描述。
每個租用戶自訂 (例如，特定承租人的額外資料表資料行)	可能。	可能。	可能。	複雜（因為所有租戶共享相同的表格）。
物件關聯對應 (ORM) 層 (例如 Ruby) 的目錄管理效率	有效率 (因為用戶端連線專用於租用戶)。	高效（因為客戶端連接特定於數據庫）。	效率適中。(視使用的 ORM、使用者/角色安全性模型和`search_path` 組態而定，用戶端有時會快取所有租用戶的中繼資料，進而導致資料庫連線記憶體使用量高。)	高效（因為所有租戶共享相同的表）。

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
合併租戶報告工作	重大努力。(您必須使用外部資料包裝函式 [FDW] 來合併所有租用戶中的資料，或擷取、轉換及載入 [ETL] 到另一個報表資料庫。)	重大努力。(您必須使用 FDW 將所有租用戶或 ETL 中的資料合併到另一個報表資料庫。)	適度的努力。(您可以使用聯集彙總所有結構描述中的資料。)	最小的努力。（所有租戶數據都在同一個表中，因此報告很簡單。)
用於報告的承租人特定唯讀執行個體 (例如，根據訂閱)	最低有效。(建立僅供讀取複本。)	適度的努力。(您可以使用邏輯複寫或AWS Database Migration Service [AWS DMS] 來設定。)	適度的努力。(您可以使用邏輯複寫或AWS DMS 進行設定。)	複雜（因為所有租戶共享相同的表格）。
資料隔離開	最佳。	更佳。您可以使用 PostgreSQL 角色來管理資料庫層級的權限。)	更佳。您可以使用 PostgreSQL 角色來管理結構描述層級的權限。)	更糟 因為所有承租人共用相同的資料表，因此您必須實作諸如用戶隔離的資料列層級安全性 [RLS] 等功能。)
承租人特定儲存加密金鑰	可能。每個 PostgreSQL 叢集都可以有自己的AWS Key Management Service [AWS KMS] 金鑰來進行儲存加密。)	不可能。(所有租用戶共用相同的 KMS 金鑰進行儲存加密。)	不可能。(所有租用戶共用相同的 KMS 金鑰進行儲存加密。)	不可能。(所有租用戶共用相同的 KMS 金鑰進行儲存加密。)

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
針對每個租用戶使用 AWS Identity and Access Management (IAM) 進行資料庫驗證	可能。	可能。	可能的（通過為每個模式具有單獨的 PostgreSQL 用戶）。	不可能（因為表由所有租戶共享）。
基礎架構	最高（因為沒有任何共享）。	適中。	適中。	最低
資料複製和儲存使用	所有租戶的最高彙總。PostgreSQL 系統目錄資料表以及應用程式的靜態和通用資料會在所有租用戶間重複。)	所有租戶的最高彙總。PostgreSQL 系統目錄資料表以及應用程式的靜態和通用資料會在所有租用戶間重複。)	適中。(應用程式的靜態和一般資料可以位於通用結構描述中，並可由其他租用戶存取。)	極小。（沒有數據重複。 該應用程序的靜態和通用數據可以位於相同的模式中。)
以承租人為中心的監控 (快速找出哪些租戶造成問題)	最低有效。由於每個租用戶都會分別監控，因此很容易檢查特定租用戶的活動。)	適度的努力。(因為所有承租人共用相同的實體資源，因此您必須套用其他篩選來檢查特定承租人的活動。)	適度的努力。(因為所有承租人共用相同的實體資源，因此您必須套用其他篩選來檢查特定承租人的活動。)	重大努力。因為所有租用戶共用所有資源 (包括資料表)，因此您必須使用繫結變數擷取來檢查特定 SQL 查詢所屬的租用戶。)
集中管理和健康/活動監控	大量的努力（設置中央監控和中央指揮中心）。	適度工作 (因為所有租用戶共用相同的執行個體)。	適度工作 (因為所有租用戶共用相同的執行個體)。	最小的努力 (因為所有租用戶共用相同的資源，包括結構描述)。

	筒倉	橋接與單獨的數據庫	具有不同綱要的橋接	游泳池
物件識別碼 (OID) 和交易識別碼 (XID) 環繞式的機會	極小。	高。(因為 OID，XID 是單一 PostgreSQL 叢集範圍的計數器，而且可能會有在實體資料庫之間有效清除的問題)。	適中。（因為 OID，XID 是一個單一的 PostgreSQL 集群範圍內的計數器）。	高。例如，單一資料表可以達到 40 億 TOAST OID 上限，視 out-of-line 資料欄數而定。）

資料層級的安全建議

需要列層級安全性 (RLS) 才能在 PostgreSQL 的集區模型中維護租用戶資料隔離。RLS 會在資料庫層級集中執行隔離原則，並消除了與軟體開發人員保持隔離的負擔。實作 RLS 最常見的方式是在 PostgreSQL 資料庫管理系統中啟用此功能。RLS 涉及過濾基於指定列中的值數據行的訪問。您可以使用兩種方法來篩選對資料的存取：

- 資料表中的指定資料欄會與目前 PostgreSQL 使用者的值進行比較。該使用者可以存取與登入 PostgreSQL 使用者相同的資料行中的值。
- 資料表中指定的資料欄會與應用程式所設定的執行階段變數值進行比較。在該工作階段期間，可存取資料行中等同於執行階段變數的值。

建議使用第二個選項，因為第一個選項需要為每個租用戶建立新的 PostgreSQL 使用者。相反地，使用 PostgreSQL 的 SaaS 應用程式在查詢 PostgreSQL 時，應負責在執行階段設定承租人特定內容。這將具有執行 RLS 的效果。您也可以在 table-by-table 基礎上啟用 RLS。根據最佳實務是，您應該為包含租用資料的所有資料的資料中的資料。

以下範例將建立兩個資料，並啟用 RLS。此範例會比較資料欄與執行階段變數的值`app.current_tenant`。

```
-- Create a table for our tenants with indexes on the primary key and the tenant’s name
CREATE TABLE tenant (
    tenant_id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,
    name VARCHAR(255) UNIQUE,
    status VARCHAR(64) CHECK (status IN ('active', 'suspended', 'disabled')),
    tier VARCHAR(64) CHECK (tier IN ('gold', 'silver', 'bronze'))
);

-- Create a table for users of a tenant
CREATE TABLE tenant_user (
    user_id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,
    tenant_id UUID NOT NULL REFERENCES tenant (tenant_id) ON DELETE RESTRICT,
    email VARCHAR(255) NOT NULL UNIQUE,
    given_name VARCHAR(255) NOT NULL CHECK (given_name <> ''),
    family_name VARCHAR(255) NOT NULL CHECK (family_name <> '')
);

-- Turn on RLS
ALTER TABLE tenant ENABLE ROW LEVEL SECURITY;
```

```
-- Restrict read and write actions so tenants can only see their rows
-- Cast the UUID value in tenant_id to match the type current_setting
-- This policy implies a WITH CHECK that matches the USING clause
CREATE POLICY tenant_isolation_policy ON tenant
USING (tenant_id = current_setting('app.current_tenant')::UUID);

-- And do the same for the tenant users
ALTER TABLE tenant_user ENABLE ROW LEVEL SECURITY;

CREATE POLICY tenant_user_isolation_policy ON tenant_user
USING (tenant_id = current_setting('app.current_tenant')::UUID);
```

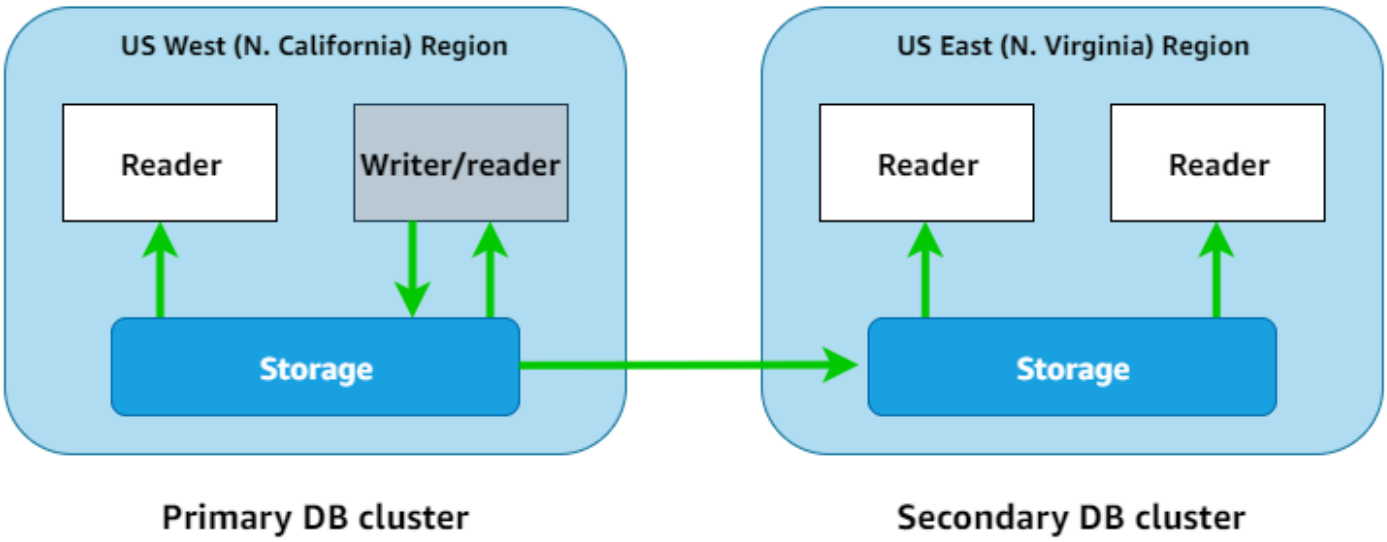
如需詳細資訊，請參閱部落格文章：使用 PostgreSQL 資料層級的安全資料層級的安全。AWSSaaS 工廠團隊也有一些示例 GitHub來協助實施 RLS。

集區模型的 PostgreSQL 可用性

集區模型本質上只有一個 PostgreSQL 執行個體。因此，針對高可用性設計應用程式至關重要。集區資料庫失敗或中斷會導致您的應用程式降級或變得無法存取所有租用戶。

Amazon RDS 版 PostgreSQL 資料庫執行個體可透過啟用高可用性功能，在兩個可用區域中設定為備援。如需詳細資訊，請參閱 Amazon RDS 文件中的適用於 Amazon RDS 的高可用性 (異地同步備份)。對於跨區域容錯移轉，您可以在不同的 AWS 區域中建立僅供讀取複本。(此僅供讀取複本必須提升為容錯移轉程序的一部分。) 此外，您還可以複寫跨 AWS 區域複寫的備份以進行復原。如需詳細資訊，請參閱 Amazon RDS 文件中的將自動備份複製到其他 AWS 區域。

Aurora PostgreSQL 相容性會自動備份資料的方式，以維持多個可用區域故障的方式。請參閱 Aurora 文件中的 Amazon Aurora 的高可用性。) 若要讓 Aurora 更具彈性並加快復原速度，您可以在其他可用區域建立 Aurora 僅供讀取複本。您可以使用 Aurora 全域資料庫將資料複寫到五個額外的區 AWS 域，以進行跨區域復原和自動容錯移轉。請參閱 Aurora 文件中的使用 Amazon Aurora 全球資料庫。) 此外，您可以使用 Aurora 全域資料庫啟用寫轉送，以實現跨多個資料庫的高可用性 AWS 區域。



無論您使用的是 Amazon RDS for PostgreSQL 版還是與 Aurora PostgreSQL 相容，我們建議您實作高可用性功能，以減輕使用集區模型的所有多租戶 SaaS 應用程式中斷的影響。

最佳實務

本節列出了本指南中的一些高級要點。有關每個點的詳細討論，請點擊相應部分的鏈接。

比較受管理 PostgreSQL 的AWS選項

AWS提供兩種在受管理環境中執行 PostgreSQL 的主要方式。（在這種情況下，託管意味著 PostgreSQL 基礎結構和數據庫管理系統由AWS服務部分或完全支持。）上的受管理 PostgreSQL 選項AWS具有自動化備份、容錯移轉、最佳化和 PostgreSQL 部分管理的優點。作為受管選項，AWS提供受管選項，提供受管選項，提供給 Amazon Reabase Service (ReAmazon Aurora RePostgreSQL Reora)。您可以透過分析 PostgreSQL 使用案例，從這兩個模型中選擇最佳選擇。如需詳細資訊，請參閱本文件中 Aurora「」一節。

選取多租用戶 SaaS 分割模式

您可以從三種適用於 PostgreSQL 的 SaaS 分割模式中進行選擇：筒倉、橋接器和集區。每個模型都有優點和缺點，您應該根據用例選擇最佳的模型。Amazon RDS for PostgreSQL stgreSQL 和 Aurora 兼容三種型號。選擇模型對於在 SaaS 應用程式中維持租用戶資料隔離至關重要。如需這些模型的詳細討論，請參閱本指南中適用於 PostgreSQL 的多租戶 SaaS 分割模型一節。

針對集區 SaaS 分割模型使用資料列層級安全性

使用 PostgreSQL 在集區模型中維護租用戶資料隔離需要資料列層級安全性 (RLS)。這是因為在集區模型中，基礎結構、PostgreSQL 資料庫或結構描述之間沒有邏輯區隔。RLS 會在資料庫層級集中執行隔離原則，並消除了與軟體開發人員保持隔離的負擔。您可以使用 RLS 將資料庫作業限制在特定承租人。如需詳細資訊和範例，請參閱本文件中的「」一節。

常見問答集

本節提供有關在多租戶 SaaS 應用程式中實作受管理 PostgreSQL 的常見問題的解答。

AWS提供哪些受管理選項？

AWS為 PostgreSQL 提供 Amazon Aurora 兼容和 Amazon Relational Database Service（亞馬遜 RDS）。 AWS也有廣泛的受管理資料庫供應項目。

哪種服務最適合 SaaS 應用程式？

您可以將 Aurora PostgreSQL 相容和 Amazon RDS for PostgreSQL SaaS 應用程式，以及本指南中討論的所有 SaaS 分割模型。這兩項服務在延展性、損毀復原、容錯移轉、儲存選項、高可用性、災難復原、備份，以及每個選項可用的執行個體類別方面都有差異。最佳選擇將取決於您的特定使用案例。使用本指南中的決策矩陣，為您的使用案例選擇最佳選項。

如果我決定將 PostgreSQL 資料庫與多租戶 SaaS 應用程式搭配使用，應該考慮哪些獨特需求？

與 SaaS 應用程式搭配使用的任何資料存放區一樣，最重要的考量是維護租用戶資料隔離的方法。如本指南所討論的，您可以透過AWS受管 PostgreSQL 產品達成租用戶資料隔離的多種方式。此外，對於任何 PostgreSQL 實作，您應該考慮以每個租用戶為基礎的效能隔離。

使用 PostgreSQL 可以使用哪些模型來維護租用戶資料隔離？

您可以使用筒倉、橋接器和集區模型做為 SaaS 分割策略，以維護租用戶資料隔離。有關這些模型以及如何將其應用於 PostgreSQL 的討論，請參閱本指南中的 PostgreSQL 的多租戶軟體 SaaS 分割模型一節。

如何使用跨多個租用戶共用的單一 PostgreSQL 資料庫來維護租用戶資料隔離？

PostgreSQL 支援資料列層級安全性 (RLS) 功能，可用來在單一 PostgreSQL 資料庫或執行個體中強制執行租用戶資料隔離。此外，您可以在單一執行個體中為每個租用戶佈建個別的 PostgreSQL 資料庫，

或是針對每個租用戶建立結構描述以達成此目標。如需這些方法的優缺點，請參閱本指南中的資料列層級安全性建議一節。

後續步驟

AWS提供兩種操作受管 PostgreSQL 的選項：Aurora 與 PostgreSQL 相容和亞馬遜 RDS 我們建議您評估這兩項服務，並針對多租戶 SaaS 應用程式選擇最能支援特定使用案例的選項。符合 SaaS 分割模型可確保使用 PostgreSQL 的 SaaS 應用程式嚴格遵守維護租用的最佳實務。SaaS 筒倉、橋接器和集區分割模型支援許多 SaaS 使用案例。這些模型在效能隔離、營運開銷和租用戶安全性等因素之間提供了不同的優勢。

後續步驟：

- 評估適用於 PostgreSQL 的 Aurora 與 Amazon RDS 相容，並為您的 SaaS 應用程式挑選最佳選項。
- 選取符合您應用程式需求的 SaaS 分割模型：筒倉、橋接器或集區。
- 根據您選取的軟體 SaaS 分割模式來實作 PostgreSQL。

資源

參考

- SaaS 儲存策略：在上建立多租戶儲存模型 AWS (AWS白皮書)
- 使用適用於 Amazon Aurora PostgreSQL 的全球資料庫進行跨區域災難復原 (AWS部落格文章)
- 使用 PostgreSQL 資料列層級安全性隔離多租戶資料 (AWS部落格文章)
- 使用 Amazon Aurora PostgreSQL (Aurora 文件)
- Amazon RDS（Amazon RDS 文檔）

合作夥伴

- 適用於 PostgreSQL 的 Amazon Aurora
- Amazon RDS for PostgreSQL 伴

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	反映 Aurora 中寫入轉送的可用性的更新。	2024年4月29 日
更新	更新了 Amazon RDS 和 Aurora 比較表。	2022 年 10 月 21 日
一	初次出版	2021 年 9 月 30 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。