

AWS Academy Cloud Architecting

模組 5：添加資料庫層



歡迎學習模組 5：添加資料庫層。

模組概覽



小節目錄

1. 架構需求
2. 數據庫層考慮因素
3. Amazon RDS
4. Amazon DynamoDB
5. 資料庫安全控制
6. 將數據遷移到 AWS 資料庫

演示

- Amazon RDS 自動備份和唯讀副本

實驗

- 指導實驗：創建 Amazon RDS 資料庫
- 挑戰實驗：將資料庫遷移到 Amazon RDS



知識測驗

本模組包含以下章節：

1. 架構需求
2. 數據庫層考慮因素
3. Amazon RDS
4. Amazon DynamoDB
5. 資料庫安全控制
6. 將數據遷移到 AWS 資料庫

本模組還包括：

- Amazon Relational Database Service (Amazon RDS) 自動備份和唯讀副本配置的演示。
- 一個指導實驗，您可以創建一個 Amazon RDS 資料庫，並使用一個簡單的 Web 應用程式連接到該資料庫。
- 一個挑戰實驗，挑戰您將數據從在 Amazon Elastic Compute Cloud (Amazon EC2) 實例上運行的資料庫遷移到 Amazon RDS 資料庫。

最後，您需要完成一個知識測驗，以測試您對本模組中涵蓋的關鍵概念的理解程度。

模組目標



學完本模組後，您應該能夠：

- 比較資料庫類型
- 區分託管服務和非託管服務
- 說明何時使用 Amazon Relational Database Service (Amazon RDS)
- 說明何時使用 Amazon DynamoDB
- 描述可用的資料庫安全控制
- 描述如何將資料移轉到 Amazon Web Services (AWS) 資料庫
- 部署資料庫伺服器

學完本模組後，您應該能夠：

- 比較資料庫類型
- 區分託管服務和非託管服務
- 說明何時使用 Amazon Relational Database Service (Amazon RDS)
- 說明何時使用 Amazon DynamoDB
- 描述可用的資料庫安全控制
- 描述如何將資料移轉到 Amazon Web Services (AWS) 資料庫
- 部署資料庫伺服器

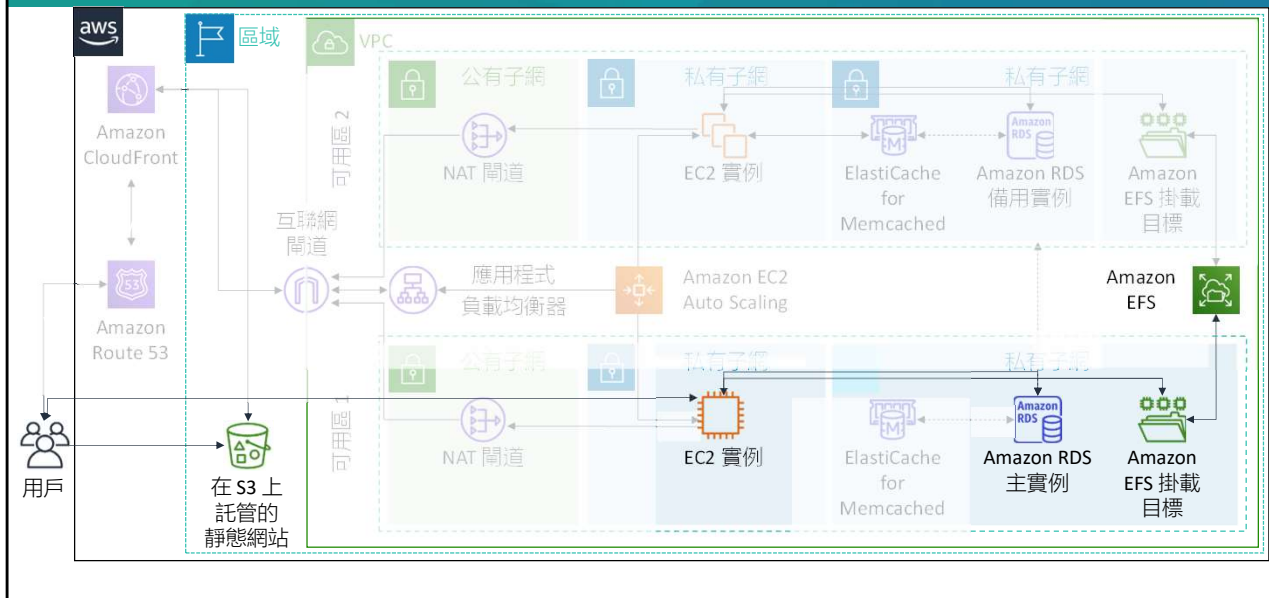
模組 5：添加資料庫層

第 1 節：架構需求



介紹第 1 節：架構需求。

資料庫是更大架構的一部分



在每個模組介紹新功能時，將逐一討論這張圖表中的各個部分。

在本模組中，您將學習如何創建使用 AWS 資料庫服務的架構。其中討論了不同的關聯式資料庫和非關聯式資料庫服務選項，包括 Amazon RDS。在瞭解到各種選項的基本優勢後，您將能夠針對資料庫服務做出更明智的決策，選擇最能滿足您的特定架構需求的服務。

咖啡館業務要求



該咖啡館需要一種更易於維護的資料庫解決方案，並應能夠提供持久性、可擴展性和高性能等基本功能。



自從該咖啡館在其網站上增加網上訂餐的功能後，咖啡館員工就注意到業務量增加了。此外，他們還發現，存儲在資料庫（該資料庫安裝在運行 Web 伺服器的同一 EC2 實例上）中的訂單歷史記錄提供了有價值的業務資訊。Martha 用它來記帳，Frank 偶爾會查看它，以便瞭解每一種甜點應該烘焙多少。

不過，Sofia 有一些擔心。資料庫需要升級和修補，而她不一定有時間定期做這些工作。此外，管理資料庫是一項專業技能，而她並不想花時間培訓其他人來完成這項工作。同時，她還擔心咖啡館沒有按應有的頻率進行資料備份。

咖啡館員工希望減少技術學習投入相關的人力成本，而這些是他們自己管理資料庫所必需的。因此，他們決定，使用託管資料庫解決方案是有必要的。理想情況下，他們將找到一種提供持久性、可擴展性和高性能等基本功能的解決方案。

在本模組中，您將詳細瞭解 AWS 提供的不同資料庫服務，以及這些服務所提供的功能。在瞭解了這些可用選項後，您應該能夠選擇一種能夠成功滿足這些新業務要求的資料庫解決方案。

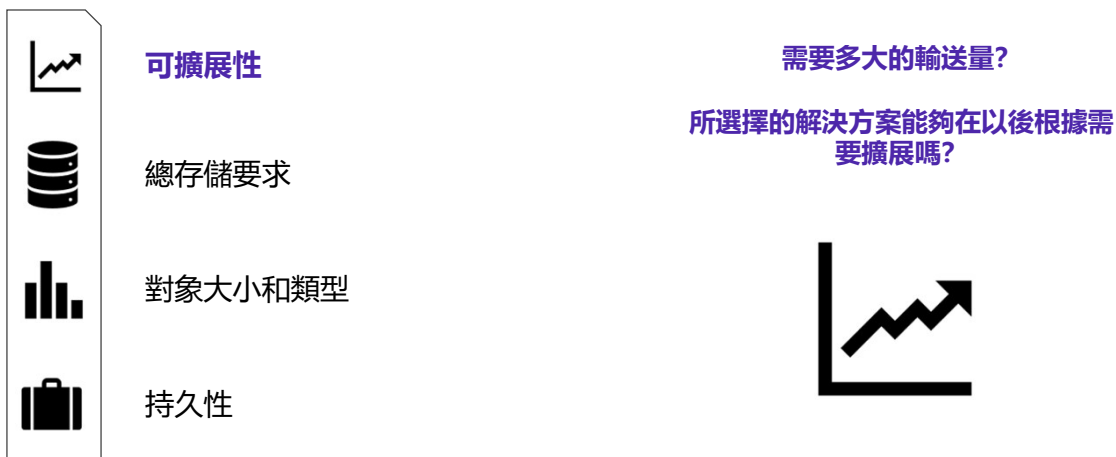
模組 5：添加資料庫層

第 2 節：資料庫層考慮因素



介紹第 2 節：資料庫層考慮因素。

資料庫考慮因素：可擴展性



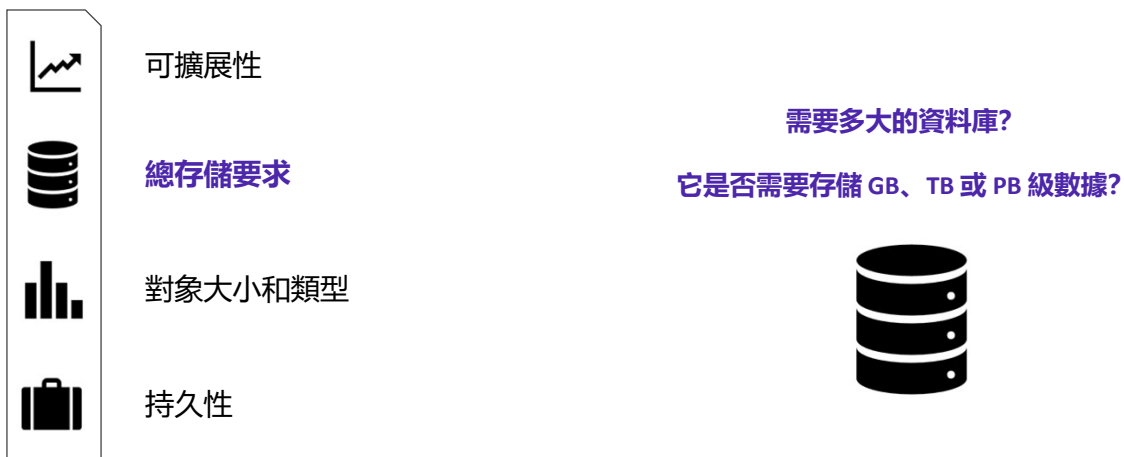
作為一名架構師，當您考慮哪種類型最適合處理特定的工作負載時，通常需要在不同的資料庫類型之間做出選擇。在選擇資料庫之前，有一些重要的考慮因素可為您的決策過程提供參考。

首先，考慮**可擴展性**的重要性。使用傳統的本地資料庫，即使對於經驗豐富的資料庫管理員來說，擴展容量也是十分艱難的任務。這可能需要幾個小時、幾天或幾周的時間。在擴展過程中，對資料庫性能的影響是不可預測的，而且可能需要停機。然而，適當擴展資料庫的重要性怎麼強調也不為過。如果資料庫預置不足，應用程式可能會停止工作。但是，如果過度預置資料庫，就會因採購了不需要的資源而增加前期成本，這違背了 AWS 架構完善的框架的成本優化原則。

理想的情況是，您可以選擇一種資料庫解決方案，該解決方案具備處理啟動時所需**輸送量**的資源，並且以後能隨您的輸送量需求增加而輕鬆**擴展**。

另一個值得擁有的功能是以後能隨您的輸送量需求或資料庫負載減少而縮減預置的資料庫容量。這樣，您通過降低預置容量便可立即實現成本節省。自動擴展解決方案可幫助降低成本和人力開銷。

資料庫考慮因素：存儲需求



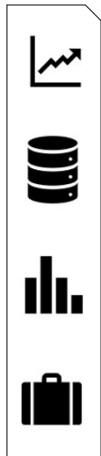
第二，當您必須選擇一個資料庫來處理特定類型的工作負載時，應考慮工作負載的存儲需求。

需要多大的資料庫來滿足您的資料需求？它是否需要存儲 GB 級、TB 級、PB 級數據？

不同的資料庫架構支援的最大資料容量不同。有些資料庫設計非常適合傳統應用，而有些則非常適合緩存或會話管理。還有一些資料庫非常適合物聯網 (IoT) 或大資料應用。

瞭解您的總存儲需求對於選擇資料庫至關重要。

資料庫考慮因素：對象大小和類型



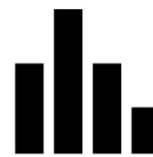
可擴展性

總存儲要求

對象大小和類型

持久性

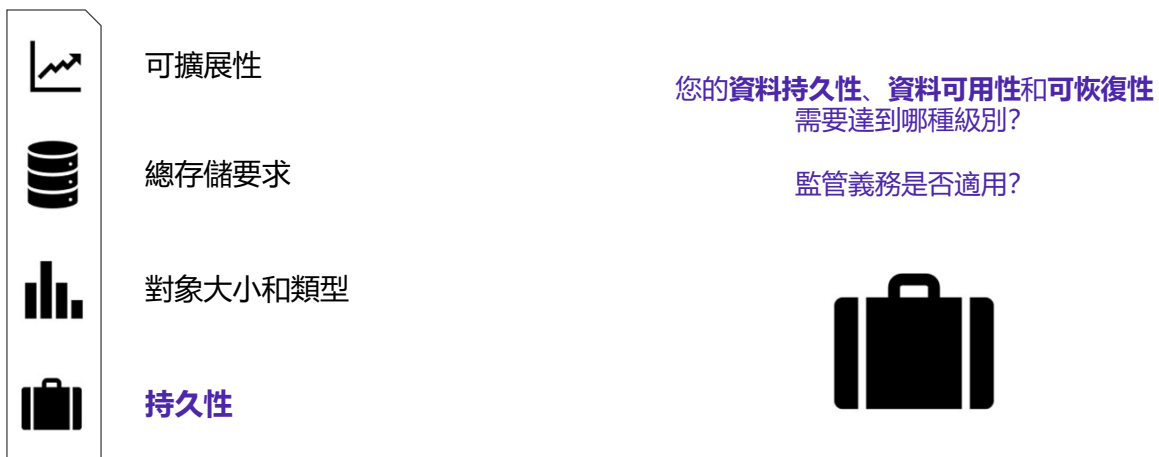
您需要存儲簡單資料結構、大型資料物件，
還是兩者都需要？



第三，當您為特定工作負載選擇資料庫時，應考慮必須存儲的物件的大小和類型。

您需要存儲簡單資料結構、大型資料物件，還是兩者都需要？

資料庫考慮因素：持久性



最後，作為您選擇資料庫的第四個考慮因素，應考慮要存儲的資料的持久性要求。

資料持久性是指保證您的資料不會丟失，**數據可用性**是指在需要時訪問資料的能力。您需要什麼級別的資料持久性和資料可用性？如果您要存儲的資料對您的業務至關重要，您應該選擇一種資料庫解決方案，將這些資料的多份冗餘副本存儲在多個地理上分離的物理位置。這種解決方案通常會導致成本增加，因此平衡業務需求和成本考慮因素很重要。

另一個重要的考慮因素是瞭解資料駐留要求或**監管義務**是否適用於您的數據。例如，是否有需要遵守的區域資料隱私法律？如果有，應選擇能夠支援合規性的資料庫解決方案。

現在您已經回顧了關鍵的考慮因素，請思考以下兩類可用的資料庫選項：

關係

傳統示例：

Microsoft SQL Server
Oracle Database
MySQL

非關係

傳統示例：

MongoDB
Cassandra
Redis

有多種類型的資料庫可供選擇，其中許多是專用的資料庫。不過，資料庫類型通常分為兩大類：關聯式資料庫和非關聯式資料庫。

關聯式資料庫系統是大多數人最熟悉的資料庫類型。傳統示例包括 Microsoft SQL Server、Oracle 資料庫和 MySQL。

非關聯式資料庫是最近發展起來的，但已經有幾十年的歷史。這類資料庫在現代計算領域發揮著至關重要的作用。示例包括 MongoDB、Cassandra 和 Redis。

關聯式資料庫類型

優勢：

- 易用性
- 資料完整性
- 減少了資料存儲
- 通用語言（結構化查詢語言或 SQL）



關聯式資料庫管理系統 (RDBMS)

在以下情況下，關聯式資料庫是理想之選：

- 需要遵守嚴格的 schema 規則、ACID 合規性和資料品質強制執行規則。
- 不需要極大的讀/寫容量
- 不需要極高的性能
 - RDBMS 可能是最適合、最省事的解決方案



關聯式資料庫有時稱為**關係資料庫管理系統 (RDBMS)**。這類資料庫仍然是世界上最受歡迎和最常用的資料庫類別。根據 2019 年的 [DB-Engines 排名研究](#)，關聯式資料庫在所有類別中人氣指數占 75% 以上。排名前四位的分別是 Oracle、MySQL、Microsoft SQL Server 和 PostgreSQL，它們都是關聯式資料庫。

關聯式資料庫一直如此受歡迎，原因有很多。其中包括**易用性**、資料完整性控制、在降低整體資料存儲方面表現出色，以及對結構化查詢語言 (SQL) 的支援。SQL 是一種用於查詢或與存儲在 RDBMS 中的結構化資料交互的流行語言，受到大多數供應商的支持。

如果您的使用案例適合**嚴格的 schema 規則**，且 schema 結構定義明確，不需要經常改變，那麼關聯式資料庫是不錯的選擇。此外，關聯式資料庫事務滿足 **ACID 合規性**，也就是說，這類資料庫通過提供遵循**原子性**、**一致性**、**隔離性**、**持久性**的事務來確保資料完整性。但是，如果您的應用程式需要極高的讀/寫容量，則關聯式資料庫可能不是合適的選擇。最後，將關聯式資料庫與其他選項進行比較時，極高性能是關聯式資料庫的一個特徵。但是，對於許多使用案例來說，關聯式資料庫可能是最省事的解決方案。

非關聯式資料庫類型



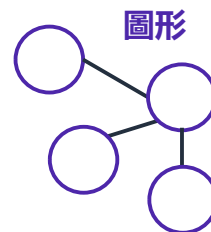
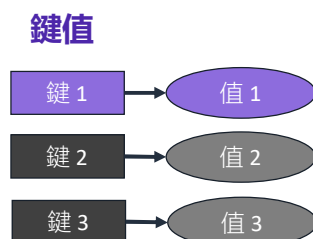
優勢

- 靈活性
- 可擴展性
- 高性能
- 功能強大的 API

在以下情況下，非關聯式資料庫是理想之選：

- 數據庫必須**橫向**擴展以便處理龐大的資料量
- 數據不適合傳統 schema
- 傳統 RDBMS 無法以經濟高效的方式支援您的讀/寫速率

示例模型



非關聯式資料庫有時稱為 *NoSQL 資料庫*。這類數據庫不同於關係資料庫，因為它們專用於特定的資料模型，並且具有靈活的 schema 來構建現代應用程式。非關聯式資料庫因其**靈活性**、**可擴展性**、**高性能**和**功能強大的 API** 而廣受讚譽。

這類資料庫使用各種資料模型，其中包括鍵值、圖形、文檔、記憶體中和搜索。NoSQL schema 是動態的。例如，行不需要在每列中都包含資料。此外，NoSQL 資料庫可以通過增加其運行的伺服器數量來**橫向**擴展。

非關聯式資料庫的 schema **靈活**，其中每個物件可以具有不同的結構。它們可以存儲結構化資料（如關聯式資料庫記錄）、半結構化資料（如 JavaScript 物件標記法 (JSON) 文檔），以及非結構化資料（如照片檔或電子郵件）。當資料可能存在結構上的不一致性時，非關聯式資料庫可以很好地發揮作用。

最後，非關聯式資料庫針對特定的資料模型和訪問模式進行了優化，可實現**更高的性能**，而不是試圖實現關聯式資料庫的功能。

存在更多的資料庫選項，這些選項是常見的示例

關聯式資料庫



Amazon RDS



Amazon Redshift



Amazon Aurora

非關聯式資料庫



Amazon DynamoDB



Amazon ElastiCache



Amazon Neptune

本模組的關注點

您已經瞭解了關聯式資料庫和非關聯式資料庫之間的區別，現在可以思考一下有哪些 Amazon 資料庫產品符合這兩種類別。

較常用的 AWS 關係資料庫產品包括 Amazon RDS、Amazon Redshift 和 Amazon Aurora。

類似地，較常用的非關係數據庫包括 Amazon DynamoDB、Amazon ElastiCache 和 Amazon Neptune。

在本模組中，您將更深入地瞭解 Amazon RDS 和 Amazon DynamoDB。有關 AWS 上提供的資料庫服務的完整清單及其最常見的使用案例，請參見 [AWS 資料庫 – 概覽頁面](#)。

第 2 節要點



- 在選擇資料庫時，應考慮可擴展性、存儲需求、要存儲的對象的類型和大小，以及持久性要求
- 關聯式資料庫具有嚴格的 schema 規則，可確保資料完整性，並支援 SQL
- 非關聯式資料庫可橫向擴展，提供更高的可擴展性和靈活性，並且很適合半結構化和非結構化資料

本模組中這節內容的要點包括：

- 在選擇資料庫時，應考慮可擴展性、存儲需求、要存儲的物件的類型和大小，以及持久性要求
- 關係資料庫具有嚴格的 schema 規則，可確保資料完整性，並支援 SQL
- 非關聯式資料庫可橫向擴展，提供更高的可擴展性和靈活性，並且很適合半結構化和非結構化資料

模組 5：添加資料庫層

第 3 節：Amazon RDS



介紹第 3 節：Amazon RDS。

關係



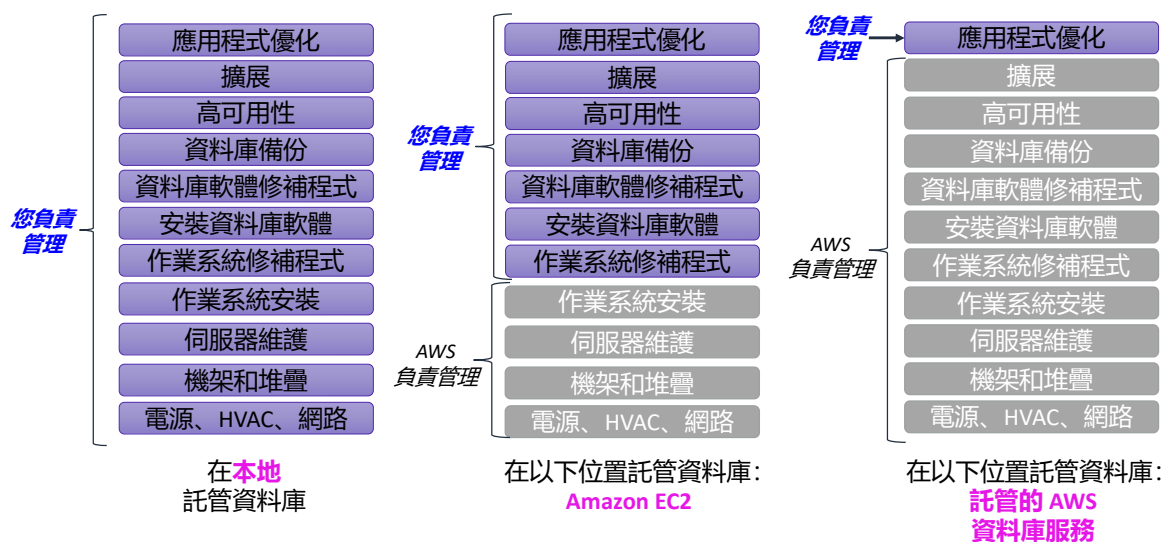
Amazon
RDS

Amazon RDS 是一項**完全託管**的關聯式資料庫**服務**。



Amazon RDS 是一項完全託管的關聯式資料庫服務，可在雲中創建和運行關聯式資料庫。在瞭解更多關於 Amazon RDS 的詳細資訊之前，您將首先瞭解 Amazon RDS 作為**托管**資料庫服務的優勢。

託管的 AWS 資料庫服務的優勢



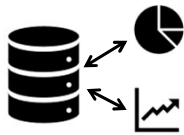
現在，您將瞭解使用託管的 AWS 資料庫服務的一些優勢。

在左側的用例中，您將資料庫託管在本地資料中心。在這種情況下，一切事情都由您負責。您必須為物理伺服器的機架供電，維護運行資料庫的伺服器的作業系統，並執行資料庫軟體的安裝和打補丁工作。您還必須執行備份，配置具備高可用性和擴展性的解決方案，並優化使用資料庫的應用程式。

在中間的用例中，您在一個或多個 Amazon EC2 示例上安裝資料庫。在這種情況下，AWS 負責物理資料中心環境的維護，並且作業系統預先安裝在您啟動的 EC2 實例上。但是，您仍然要負責作業系統安裝之上的每一個配置層，這意味著您必須手動管理很多資源。

右側的使用案例使用的是 AWS 託管資料庫產品。這些解決方案提供高可用性、可擴展性和資料庫備份作為您可以配置的內置選項。AWS 負責處理常見的、重複性的資料庫管理任務。您只負責優化您的應用程式，並確保資料庫層有效地適用於您的應用程式。

Amazon RDS 特性



訪問模式
事務性羽量級分析



數據大小
低 TB 範圍



性能
中高輸送量、低延遲



業務使用案例
事務性
OLAP

Amazon Relational Database Service (Amazon RDS) 是一項完全託管的關聯式資料庫服務。

Amazon RDS 通常用於事務性訪問模式的使用案例，或用於羽量級分析。

作為關聯式資料庫，理想的資料大小範圍可達到低 TB 範圍。隨著存儲需求的增長，您可以預置額外的存儲。Amazon Aurora 引擎會隨著您資料庫存儲需求的增長而自動增加您的資料庫容量大小。它最多可擴展至 64TB 或您定義的最大容量。MySQL、MariaDB、Oracle 和 PostgreSQL 引擎使您能夠擴展到 32TB 的存儲空間。Microsoft SQL Server 最高支持 16TB。存儲擴展是動態的，無停機時間。

在性能方面，Amazon RDS 提供了兩種選項。其中一個是通用固態硬碟 (SSD) 支援的存儲選項。SSD 選項提供了一致的基準，即預置的每 GB 有 3 IOPS，並且可以突增至高出基準達 3000 IOPS。這種存儲類型適用於廣泛的資料庫工作負載。另一個是預置 IOPS SSD 存儲選項，非常適用於 I/O 密集型事務性資料庫工作負載。

Amazon RDS：用途和資料庫類型



Amazon
RDS

非常適合以下應用：

- 具有更複雜的資料
- 需要合併和聯接資料集
- 需要強制實施語法規則

支援的六種資料庫類型：

- | | |
|------------------------|--------------|
| • Microsoft SQL Server | • PostgreSQL |
| • Oracle | • Aurora |
| • MySQL | • MariaDB |



作為關聯式資料庫產品，Amazon RDS 對於具有複雜、結構化良好資料的應用程式來說是不錯的選擇。如果您的工作負載必須經常合併和聯接資料集，並且必須有嚴格執行的語法規則，那麼 Amazon RDS 是不錯的選擇。例如，Amazon RDS 經常用於支援傳統應用、企業資源規劃 (ERP)、客戶關係管理 (CRM) 和電子商務應用。

Amazon RDS 有六個資料庫引擎可供選擇，包括 Microsoft SQL Server、Oracle、MySQL、PostgreSQL、Amazon Aurora 和 MariaDB。

資料庫實例大小調整



	T 系列	M 系列	R 系列
類型	突增型 實例	通用型 實例	記憶體優化型 實例
調整大小	1 個 vCPU/1GB RAM 至 8 個 vCPU/32GB RAM	2 個 vCPU/8GB RAM 至 96 個 vCPU/384GB RAM	2 個 vCPU/16GB RAM 至 96 個 vCPU/768GB RAM
聯網	中等性能	高性能	高性能
適合的工作 工作負載	小型或可變	CPU 密集型	查詢密集型、 連接次數高
亮點	T3 可突增到基準以上， 需額外付費	M5 提供多達 96 個 vCPU	R5 提供多達 96 個 vCPU、768GiB RAM

所有類型的 Amazon RDS 資料庫都運行在服務上，但 Aurora 是個例外，它可以作為無伺服器選項運行。Amazon RDS 可用於多種資料庫實例類型，這些實例類型針對不同類型的工作負載進行了優化。

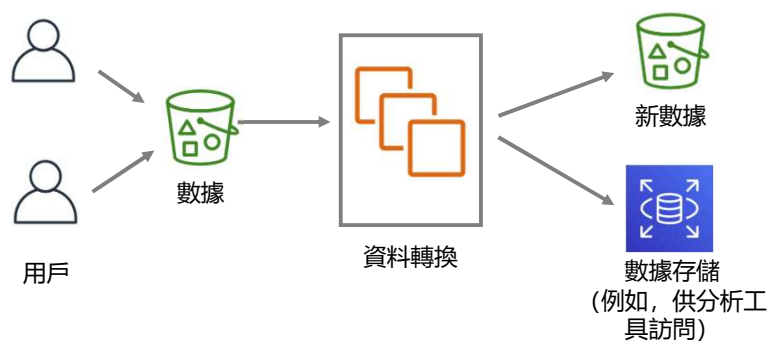
T 系列實例類型提供基準水準的 CPU 性能，並且只要需要，隨時能夠突增 CPU 使用率。例如，T3 實例可均衡計算、記憶體和網路資源，非常適用於 CPU 使用率中等但使用量可能臨時激增的資料庫工作負載。

M 系列實例是另一種通用選項，但 M 系列為 CPU 密集型工作負載提供了更多的選擇。對於開源或企業應用中的中小型資料庫，M 實例是不錯的選擇。

最後，R 系列實例針對記憶體密集型資料庫工作負載進行了優化。

部分資料庫引擎支援其他的資料庫實例類。有關更多資訊，請參閱 [Amazon RDS 資料庫實例](#)，特別是參閱 AWS 文檔中的[選擇數據庫實例類](#)頁面，瞭解實例類的詳細資訊。

分析

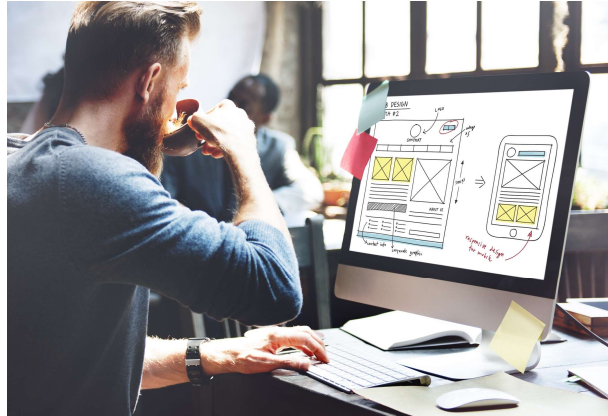


試想這樣一個示例使用案例，其中 Amazon RDS 用作部署的業務挑戰解決方案的一部分。

在本例中，一家公司希望從他們的一些業務資料中獲得有意義的見解。使用者將資料上傳到 Amazon Simple Storage Service (Amazon S3) 存儲桶。然後，使用 Amazon EC2 實例集群來轉換這些資料。這可能以批次處理的形式定期進行。轉換後的資料將存儲在另一個 S3 存儲桶中。但是，部分轉換後的資料會被插入到一個 RDS 實例中，該實例標記為數據存儲。之後可以使用分析軟體或 AWS 分析服務查詢此 RDS 實例，這些分析軟體或服務可通過 SQL 查詢來訪問資料，獲得業務見解。

模組 5 – 指導實驗： 創建 Amazon RDS 資料庫

aws academy

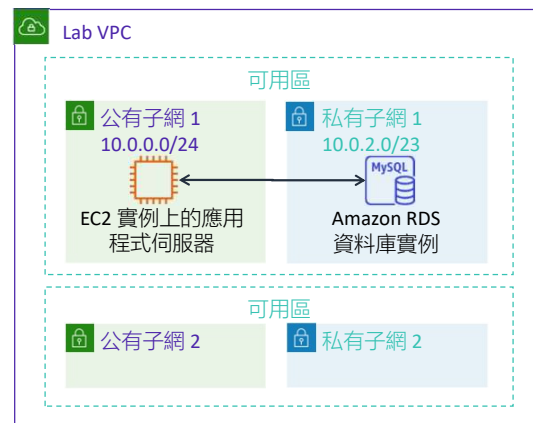


您現在將完成模塊 5 – 指導實驗：創建 Amazon RDS 資料庫。

指導實驗：任務



1. 創建 Amazon RDS 資料庫
2. 配置 Web 應用程式與資料庫實例的通信



在本指導實驗中，您將完成以下任務：

1. 創建 Amazon RDS 資料庫
2. 配置 Web 應用程式與資料庫實例的通信



大約 20 分鐘



開始模組 5 – 指導實驗：創建 Amazon RDS 資料庫

現在可以開始指導實驗了。

指導實驗總結： 要點



完成這個指導實驗之後，您的講師可能會帶您討論此指導實驗的要點。

通過多可用區部署實現高可用性

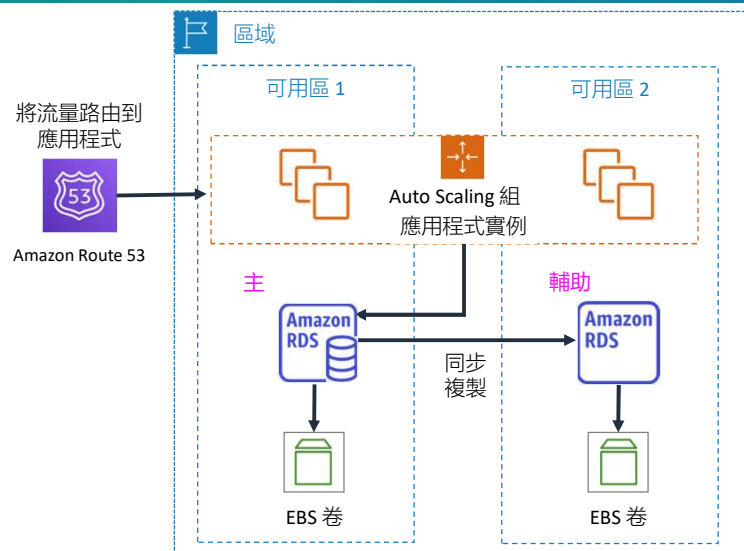


優勢

- 提高持久性
- 提高可用性
- 自動容錯移轉到備用可用區

自動執行容錯移轉的條件

- 主可用區失去可用性
- 與主資料庫的網路連接斷開
- 主實例的計算單位出現故障
- 主實例出現存儲故障



Amazon RDS 通過實施多可用區方法提供高可用性。Amazon RDS 會自動在不同可用區中預置和維護同步備用實例。主要資料庫實例將跨可用區同步複製到備用實例，以提供資料冗餘、消除 I/O 凍結，並在系統備份期間將延遲峰值降至最小。

運行具有高可用性的資料庫實例可提高計畫內系統維護期間的可用性。Amazon RDS 可說明保護資料庫免受資料庫實例故障和可用區中斷的影響。它還能提高資料庫實例的可用性和持久性，使其成為生產資料庫工作負載的不錯選擇。

如果基礎設施出現故障，Amazon RDS 會將故障自動轉移到備用實例。容錯移轉完成後，您可以恢復資料庫操作。容錯移轉後資料庫實例的終端節點維持不變，因此您的應用程式可在沒有手動管理干預的情況下恢復資料庫操作。

容錯移轉條件包括主可用區失去可用性、與主要資料庫的網路連接斷開、主實例上的計算單位故障或存儲故障。

使用唯讀副本提高性能



優勢

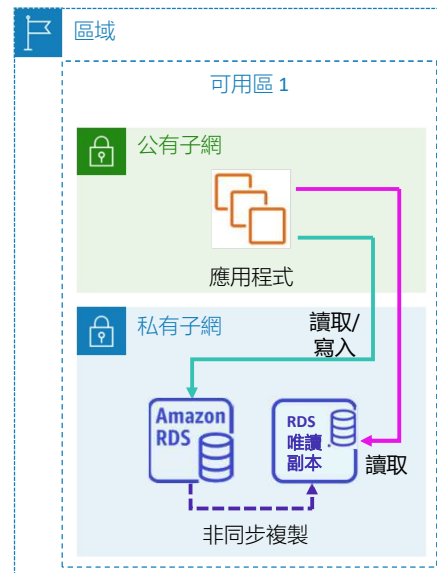
- 提高性能
- 提高可用性
- 專為安全性設計

受以下資料庫支援

- MySQL
- MariaDB
- PostgreSQL
- Oracle

限制

- 每個資料庫最多可以有五個唯讀副本
- 為了實現嚴格的先寫後讀一致性，需從主要資料庫實例讀取

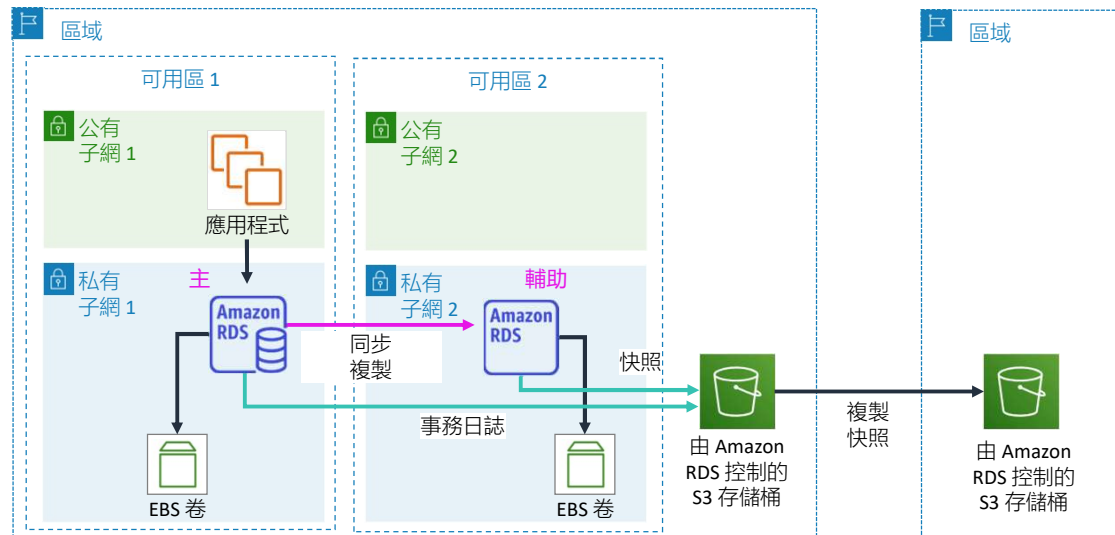


Amazon RDS 使您能夠使用來源資料庫實例創建一種稱為*只讀副本*的特殊類型的資料庫實例。對源資料庫實例的更新將非同步複製到唯讀副本。

使用唯讀副本可以提高*性能*。例如，您可以通過將唯讀查詢從應用程式路由到唯讀副本來減少來源資料庫實例的負載。唯讀副本還可以提高*可用性*。對於讀取量較大的資料庫工作負載，您可以進行彈性擴展，超越單個資料庫實例的容量限制。如有需要，您也可以手動將唯讀副本提升為獨立的資料庫實例。

Amazon RDS for MySQL、MariaDB、PostgreSQL 和 Oracle 均支持唯讀副本。這些資料庫引擎中的每一個支援每個主要資料庫最多有五個唯讀副本。如果您需要嚴格的先寫後讀一致性（也就是說，*您讀取的內容始終是剛寫入的內容*，即使是立即讀取），則應從主要資料庫實例中讀取。否則，您可以分配負載，並從唯讀副本中讀取。

Amazon RDS 備份解決方案



現在，試想這樣一個基於 Amazon RDS 的備份解決方案的示例架構。

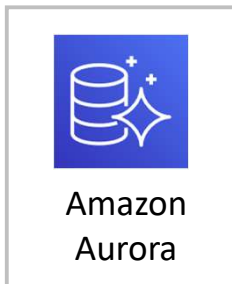
在本例中，RDS 資料庫的快照和事務日誌存儲在由 Amazon RDS 控制的 S3 存儲桶中。借助 Amazon RDS，您還可以將其指定為將自動或手動資料庫快照複製到另一個 AWS 區域。您也可以將快照複製到單獨的 AWS 帳戶。

有關更多資訊，請參閱 AWS 文檔中的[複製快照](#)。

演示：
Amazon RDS 自動備
份和唯讀副本



現在，講師可以選擇使用 AWS 管理主控台來演示在 Amazon RDS 上配置自動備份和唯讀副本。



Amazon Aurora 是完全託管且與 MySQL 和 PostgreSQL 相容的關聯式資料庫引擎。

- 用於連線交易處理 (OLTP)
- 最多可將 MySQL 輸送量增加至五倍*
- 最多可將 PostgreSQL 輸送量增加至三倍*
- 跨三個可用區以六條路徑複製資料
- 只需對您現有的應用程式進行少量更改

*以下資料庫提供詳細的基準測試資訊：[MySQL](#) 和 [PostgreSQL](#)。

Amazon Aurora 是 Amazon RDS 的一種資料庫引擎選項。

Aurora 是專為雲構建與 MySQL 和 PostgreSQL 相容的關聯式資料庫。它既具有傳統企業級資料庫的性能和可用性，又具有開來源資料庫的簡單性和成本效益。

使用 Aurora 的速度最高可達標準 MySQL 資料庫的五倍，標準 PostgreSQL 資料庫的三倍。Aurora 可以實現商用資料庫的安全性、可用性和可靠性，而成本只有商用資料庫的 1/10。有關此基準的詳細說明以及如何自行複製此基準，請參閱 [Amazon Aurora MySQL 性能基準指南](#) 和 [Amazon Aurora PostgreSQL 性能基準指南](#)。

Aurora 採用一種具有容錯能力並能自我修復的分散式存儲系統，這一系統可以把每個資料庫實例自動擴展到最高 64TB。它具備高性能和高可用性，支援最多 15 個低延遲唯讀副本、時間點恢復、持續備份到 Amazon S3，還支持跨三個可用區複製。

Aurora 常用於 **聯機事務處理** (OLTP)。OLTP 系統必須能夠處理大量併發用戶的操作，並能夠運行插入和更新請求。一個常見的 OLTP 示例是訂單錄入系統。OLTP 通常與 **聯機分析處理** (OLAP) 形成對比，後者的特點是更複雜的查詢量較少。



Amazon Redshift 是一項數據倉庫服務。

- 用於連線分析處理 (OLAP)
- 存儲非常大的資料集
 - 在 Amazon Redshift 中存儲高度結構化、頻繁訪問的資料
 - 此外，可以在 Amazon S3 中存儲 EB 級結構化、半結構化和非結構化資料。

Amazon Redshift 是另一款 AWS 關聯式資料庫產品，它不在 Amazon RDS 上運行。

Amazon Redshift 提供 PB 級資料倉庫和資料湖分析，Amazon Redshift 通常用於存儲頻繁訪問、高度結構化的資料。Amazon Redshift 還可以直接訪問 Amazon S3 中的資料，因此您可以在 Amazon S3 中維護 EB 級的結構化、半結構化和非結構化資料。

即使有數千個併發查詢，Amazon Redshift 也能提供始終如一的快速性能。無論您要查詢 Amazon Redshift 資料倉庫中的資料，還是查詢 Amazon S3 資料湖中的資料，都可獲得一致的性能。您可以查詢 Parquet、JSON、Avro、CSV 等開放檔案格式，也可以直接使用 SQL 查詢 Amazon S3。

第 3 節要點



- 託管的 AWS 資料庫服務負責處理管理任務，讓您可以專注于自己的應用程式
- Amazon RDS 支持 Microsoft SQL Server、Oracle、MySQL、PostgreSQL、Aurora 和 MariaDB
- Amazon RDS 多可用區部署通過自動容錯移轉提供高可用性
- 每個主要資料庫最多可以有五個只讀副本，從而提高 Amazon RDS 的性能
- Amazon Aurora 是一種完全託管且與 MySQL 和 PostgreSQL 相容的關聯式資料庫引擎
- Amazon Redshift 是一款資料倉庫關係資料庫產品

本模組中這節內容的要點包括：

- 託管的 AWS 資料庫服務負責處理管理任務，讓您可以專注于自己的應用程式
- Amazon RDS 支持 Microsoft SQL Server、Oracle、MySQL、PostgreSQL、Aurora 和 MariaDB
- Amazon RDS 多可用區部署通過自動容錯移轉提供高可用性
- 每個主要資料庫最多可以有五個唯讀副本，從而提高性能
- Amazon Aurora 是一種完全託管且與 MySQL 和 PostgreSQL 相容的關聯式資料庫引擎
- Amazon Redshift 是一款資料倉庫關聯式資料庫產品

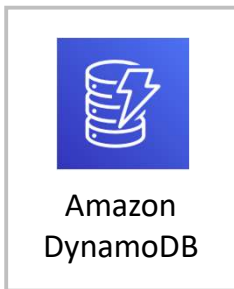
模組 5：添加資料庫層

第 4 節：Amazon DynamoDB



介紹第 4 節：Amazon DynamoDB。

Amazon DynamoDB



一項完全託管的**非關係**
鍵值和**文檔**數據庫服務。



任意規模下
均提供高性能

極強的橫向擴
展能力



無伺服器

事件驅動型程式設
計（無伺服器計算）



適合企業使用

加密、存取
控制、備份

Amazon DynamoDB 是一項完全託管的非關係 NoSQL 資料庫服務。它憑藉無縫可擴展性提供快速可預測的性能。

主要優勢如下：

- 大規模性能
 - 提供只有幾毫秒的一致響應時間
 - 您可以構建輸送量幾乎無限的應用程式
- 無伺服器
 - 無需預置、修補或管理伺服器
 - 無需安裝、維護或操作軟體
- 適合企業使用
 - 支持 ACID 事務
 - 默認情況下靜態加密所有資料
 - 提供多區域複製（全域表）
 - 提供精細的身份與存取控制
 - 您可以執行完整的資料備份，而不會影響性能



Amazon
DynamoDB

非常適合以下應用：

- 具有簡單的大容量資料（高 TB 範圍）
- 必須快速擴展
- 不需要複雜的聯接
- 需要超高輸送量和低延遲

主要功能

- NoSQL 表
- 專案可以擁有不同屬性
- 內存中的緩存
- 支持每秒超過 2000 萬個請求的峰值

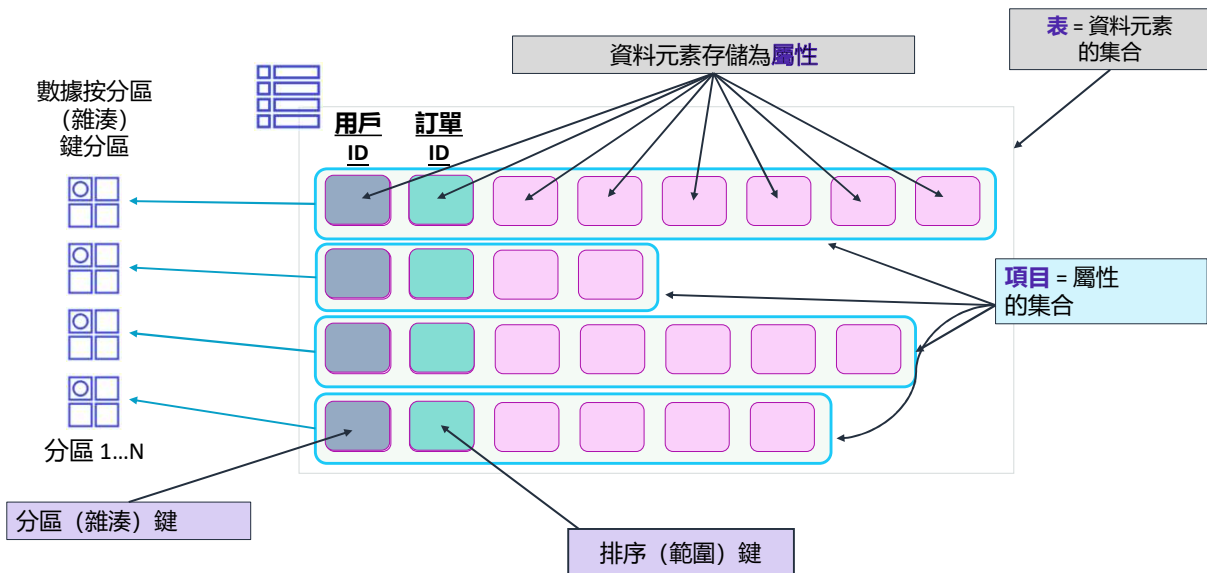


DynamoDB 非常適合要處理大量資料且必須快速擴展的應用程式。它還提供超高的輸送量和較低的延遲，對於遊戲、廣告技術、移動應用程式以及有這些要求的其他應用程式來說，是一個不錯的選擇。但是，如果您的工作負載需要複雜的聯接，關聯式資料庫可能是更好的選擇。

DynamoDB 表沒有固定的 schema，每一個項目可以擁有不同數量的屬性。

由於 DynamoDB 提供記憶體中的緩存，每秒可以處理超過 2000 萬個請求，因此經常用於必須維持會話狀態的應用程式、無伺服器 Web 應用程式和微服務。

Amazon DynamoDB 資料模型



現在來看一下 Amazon DynamoDB 資料模型。由於 DynamoDB 是 NoSQL 資料庫，因此與關聯式資料庫不同，它不強制執行嚴格的 schema。

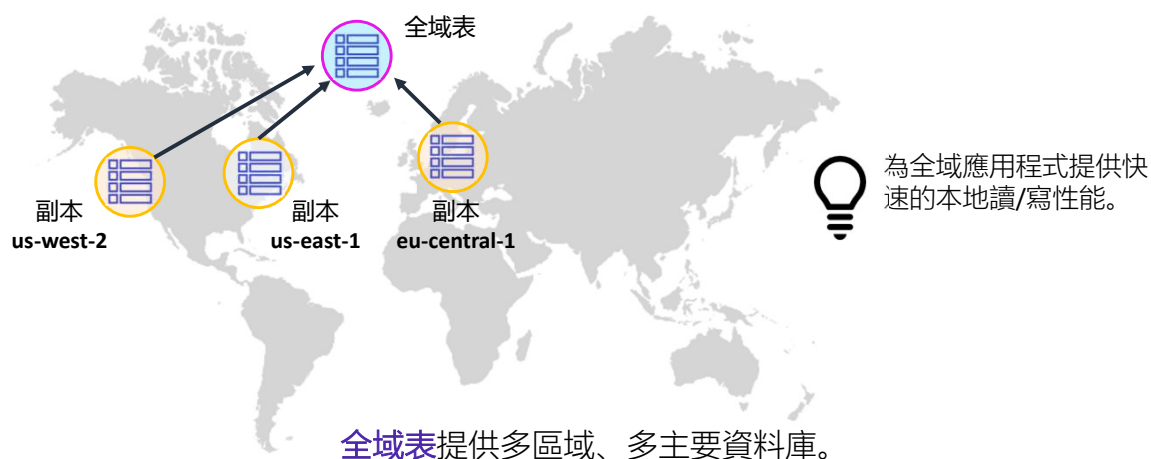
DynamoDB 表包含項目。一個項目的屬性數量不固定。每個屬性由一個鍵值對組成。

DynamoDB 支援使用用戶定義的主鍵（分區鍵，也稱為雜湊鍵）來執行鍵值 GET/PUT 操作。主鍵是表中專案必須具備的唯一屬性，可以通過它唯一地標識每個專案。您在創建表時指定主鍵。此外，DynamoDB 允許使用全域二級索引和本地二級索引對非主鍵屬性進行查詢，從而提供靈活的查詢。

當資料集的大小增加以及隨著預置的容量增加，會發生自動分區。

考慮這樣一個例子，您創建了一個 Amazon DynamoDB 表來保存訂單資訊。在本例中，您的表條目將始終具有用戶 ID 和訂單 ID。但是，您只能收集部分訂單的其他詳細資訊（如使用者的電子郵寄地址），不能收集另外一些訂單的這些資訊。隨著更多的項目添加到表中，您可以繼續收集新的屬性。這些屬性可以包括您在開始創建表時沒有預期到的屬性。由於具有這種靈活性，DynamoDB 非常適合快速變化的非結構化、半結構化和結構化資料。

Amazon DynamoDB 全域表



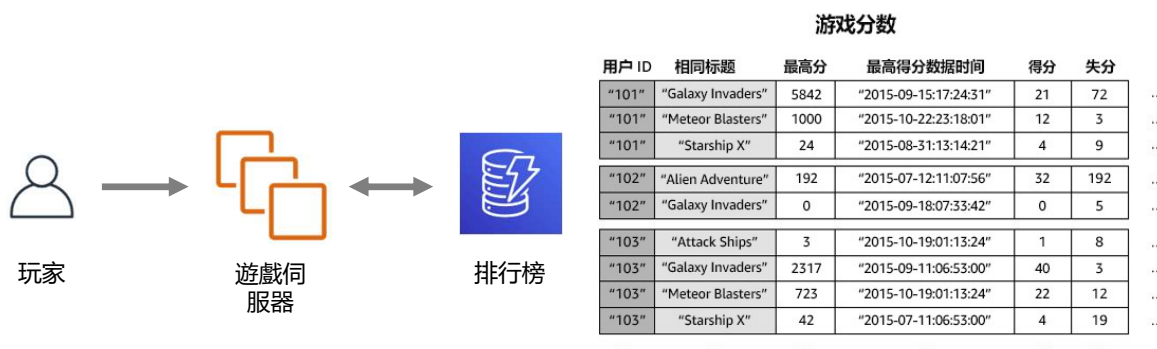
預設情況下，Amazon DynamoDB 會在一個區域中跨多個可用區複製您的資料。但在某些情況下，您可能希望跨多個區域複製資料。

Amazon DynamoDB 全域表為部署多區域、多主要資料庫提供了完全託管的解決方案。您不需要構建和維護自己的複製解決方案。在創建全域表時，指定要在其中提供表的 AWS 區域。DynamoDB 執行所有必要的任務，以便在這些區域中創建相同的表。然後，DynamoDB 會將持續的資料更改傳播到全部的表。

為說明全域表的一個使用案例，假設您有一個大型客戶群跨越三個地理區域：美國東海岸、美國西海岸和西歐。客戶在使用您的應用程式時必須更新其個人資料資訊。

為了滿足這一業務要求，您可以創建由三個特定於區域的 CustomerProfiles 表構成的全域表。然後，DynamoDB 會自動在這些表之間複製資料更改。在一個區域中對 CustomerProfiles 資料所做的更改將無縫傳播到其他區域。此外，如果其中一個 AWS 區域暫時不可用，您的客戶仍然可以訪問其他區域中的相同 CustomerProfiles 資料。

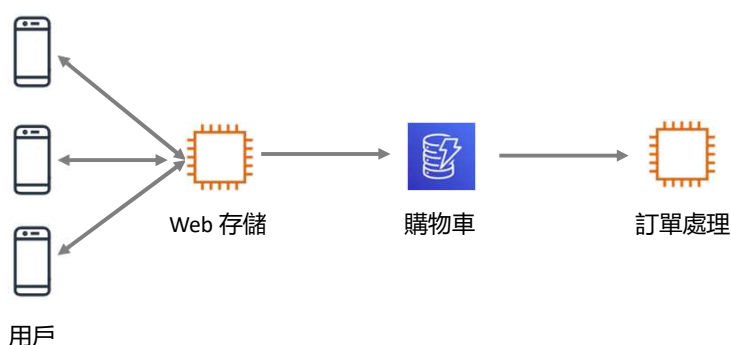
排行榜和評分



遊戲公司將 Amazon DynamoDB 應用於遊戲平臺的各個部分，包括遊戲狀態、玩家資料、會話歷史記錄和排行榜。您可以可靠地擴展到數百萬併發用戶和請求，同時確保始終保持低延遲。

在此示例使用案例中，在 Amazon DynamoDB 中維護一個遊戲得分排行榜。主動連接到遊戲伺服器的玩家可能會通過遊戲使用者介面 (UI) 看到其中的一些資訊。當新的高分出現時，GameScores 排行榜會不斷更新，每個高分玩家的勝負記錄也會發生變化。以 Electronic Arts (EA) 為例，這是一家大型視頻遊戲公司，在全球擁有超過 3 億註冊玩家。對於 EA 來說，高併發意味著每秒 10 萬次以上的請求和數百萬日活躍用戶。有關詳細資訊，請參閱 [DynamoDB 遊戲使用案例](#)。

臨時數據（線上購物車）



之所以開發 Amazon DynamoDB，是因為客戶需要一個可擴展且高度可靠的鍵值資料庫來支援 Amazon 電子商務的基本操作，如購物車。

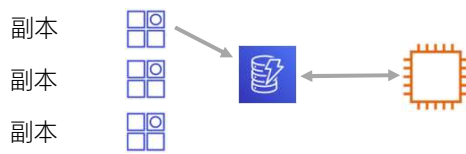
2012 年，Amazon 首席技術官 Werner Vogels 在宣佈發佈 Amazon DynamoDB 時，在一篇[博文](#)中說明了 Amazon 購物車的業務要求。“這種非關係或 NoSQL 資料庫針對購物車等使用案例打造，任何停機時間或性能下降...都會對收益產生直接影響，它們對資料系統的容錯能力和性能要求非常嚴格。這些服務還需要能夠逐步擴展基礎設施，以適應請求速率或資料集大小的增長。另一個重要的要求...是可預測性。”

DynamoDB 功能集不斷發展。但是，激勵開發這項服務的核心目標如今仍然適用於希望實施網店購物車訂單處理解決方案的企業或組織。

Amazon DynamoDB 一致性選項

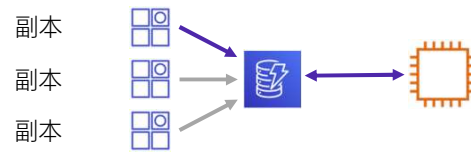


最終一致性



默認設置。所有資料副本通常在 **1 秒** 內達到一致。

強一致性



此功能為可選功能。適合需要所有讀取均返回反映讀取之前所有寫入操作結果的應用。

讀取一致性表示的是某個資料項目目的成功寫入或更新*如何*以及*何時*反映在該資料項目的後續讀取操作中。Amazon DynamoDB 提供的邏輯使您能夠針對應用程式中的每個讀取請求，指定您需要的一致性特徵。

最終一致性讀取

當您從 DynamoDB 表讀取資料時，回應反映的可能不是剛剛完成的寫入操作的結果。回應可能包含某些陳舊資料。如果您在短時間後重複讀取請求，回應將返回最新的資料。

強一致性讀取

當您請求強一致性讀取時，DynamoDB 會返回具有最新資料的回應。這些回應反映了所有之前成功寫入操作的更新。如果網路延遲或中斷，可能會無法執行強一致性讀取。

除非您指定其他讀取方式，否則 DynamoDB 將使用**最終一致性讀取**。讀取操作（如 GetItem、Query 和 Scan）提供了一個 ConsistentRead 參數。如果您將此參數設置為 *true*，DynamoDB 將在操作過程中使用強一致性讀取。

課堂討論： 咖啡館應該使 用哪個資料庫？



現在您已經瞭解了 Amazon RDS 和 Amazon DynamoDB 的基本關鍵功能，請思考在這兩種資料庫服務中，哪一種更適合咖啡館的使用案例。

在本課程前面的模組中，您完成了一項挑戰實驗，您在其中部署了在 EC2 實例上運行的 MySQL 資料庫。該資料庫託管一個表，其中包含有關咖啡館功能表的詳細資訊。它還提供資料存儲層，用來存儲客戶線上下的訂單。

如果咖啡館團隊決定他們想遷移資料庫層，以便改為使用 Amazon 託管的資料庫服務，那麼這兩個資料庫選項中的哪一個更合適？

講師將帶班級學員討論這個主題。我們鼓勵您參與討論，並解釋您給出的答案的原因。

第 4 節要點



- Amazon DynamoDB 是一項完全託管的非關係鍵值和文檔 NoSQL 資料庫服務。
- DynamoDB 是無伺服器服務，可提供極高的橫向擴展和低延遲性能。
- DynamoDB 全域表可確保資料複製到多個區域。
- DynamoDB 默認提供最終一致性（一般情況下，在寫入後的 1 秒鐘內的讀取完全一致）。強一致性是另一種選項。

本模組中這節內容的要點包括：

- Amazon DynamoDB 是一項完全託管的非關係鍵值和文檔 NoSQL 資料庫服務。
- DynamoDB 是無伺服器服務，可提供極高的橫向擴展和低延遲性能。
- DynamoDB 全域表可確保資料複製到多個區域。
- DynamoDB 預設提供最終一致性（一般情況下，在寫入後的 1 秒鐘內的讀取完全一致）。強一致性是另一種選項。

模組 5：添加資料庫層

第 5 節：資料庫安全控制



介紹第 5 節：資料庫安全控制。

保護 Amazon RDS 資料庫



建議

- 在 **Virtual Private Cloud (VPC)** 中運行 RDS 實例
 - 提供服務隔離和 IP 防火牆保護
- 使用 **AWS Identity and Access Management (IAM) 策略** 進行身份驗證和訪問
 - 許可權決定誰可以管理 Amazon RDS 資源
- 使用 **安全性群組** 來控制哪些 IP 位址或 EC2 實例可以連接到您的資料庫
 - 默認禁用網路訪問
- 使用 **安全通訊端層 (SSL)** 加密傳輸中的資料
- 在資料庫實例和快照上使用 Amazon RDS **加密**，以保護靜態資料
- 使用 **資料庫引擎的安全功能** 控制可以登錄資料庫實例上的資料庫的人員
- 配置事件通知，以便在發生重要的 Amazon RDS 事件時提醒您



Amazon RDS



安全性是您和 AWS 的共同責任。AWS 負責雲的安全性，這意味著，AWS 保護運行 Amazon RDS 的基礎設施，而客戶負責雲中的安全性。

Amazon RDS 的一項安全建議是在 *Virtual Private Cloud (VPC)* 中運行您的 RDS 實例。VPC 使您能夠將實例放置在私有子網中，確保其不受互聯網上的公有路由影響。VPC 還提供 IP 防火牆保護，使您能夠安全地控制適用的網路配置。

此外，我們建議您：

- 使用 **AWS Identity and Access Management (IAM) 策略** 進行身份驗證和控制訪問。
- 配置 **安全性群組** 來限制連接。打開可訪問資料庫的 TCP 埠。但是，您可以限制這些連接的來源。
- 使用 **SSL 連接** 確保與資料庫之間的所有通信都受到保護。
- 使用 **Amazon RDS 加密** 來加密您的資料和資料庫快照。
- 使用 **資料庫引擎的安全功能**，實現強制實施密碼複雜性等要求。

您可以啟用事件通知，以便在 RDS 實例上發生重要事件時接收提醒。這些事件可能包括實例是否已關閉、備份是否啟動、是否發生容錯移轉、安全性群組是否發生更改或者存儲空間是否不足等。

建議

- 使用 **IAM 角色** 對訪問進行身份驗證
- 使用 **IAM 策略**
 - 定義使用 DynamoDB API 的精細存取權限
 - 定義表、專案或屬性級別的存取權限
 - 遵循**授予最小許可權的原則**
- 配置 **VPC 終端節點**
 - 防止連接流量遍歷開放的互聯網
 - VPC 終端節點策略讓您能夠控制和限制對 DynamoDB 表的 API 訪問
- 考慮使用**客戶端加密**
 - 在盡可能靠近資料來源的位置加密資料



Amazon
DynamoDB

預設情況下提供的安全功能

- 對存儲在表、索引、流和備份中的所有使用者資料進行**靜態加密**
- **傳輸中加密** – 所有與 DynamoDB 和其他 AWS 資源之間的通信都使用 HTTPS

為了保護 Amazon DynamoDB 的安全，您應該用來保護 Amazon RDS 的許多相同的最佳實踐也適用。

例如，使用 **IAM 角色** 來確保安全的身份驗證，以及使用 **IAM 策略** 來定義存取權限。

如果您只需要訪問從 Virtual Private Cloud (VPC) 中的 DynamoDB，則應使用 **VPC 終端節點** 來限制為僅從所需的 VPC 訪問。這樣做可以防止流量遍歷開放的互聯網，受到該環境的影響。

此外，如果您將敏感或機密資料存儲在 DynamoDB 中，您可能希望在盡可能靠近資料來源的位置加密資料，以便在該資料的整個生命週期內對其進行保護。

請注意，預設情況下，DynamoDB 提供特定的安全功能。例如，DynamoDB 可保護靜態存儲的使用者資料、本地用戶端與 DynamoDB 之間傳輸的資料，以及 DynamoDB 與同一 AWS 區域中的其他 AWS 資源之間傳輸的資料。

有關更多詳細資訊，請閱讀 [DynamoDB 安全性最佳實踐](#) 文檔。

模組 5：添加資料庫層

第 6 節：將資料移轉到 AWS 資料庫



介紹第 6 節：將資料移轉到 AWS 資料庫。

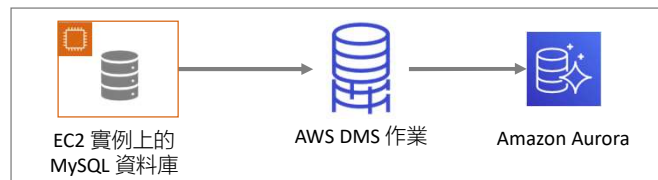
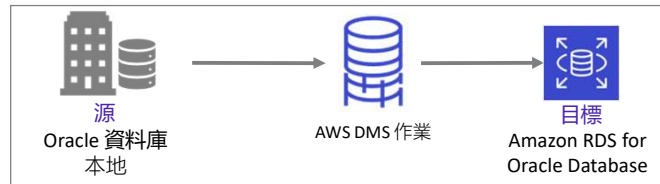
AWS Database Migration Service



AWS Database Migration Service (AWS DMS)

- 用於大多數商用和開來源資料庫之間的遷移
- 在 Amazon EC2、Amazon RDS、Amazon S3 上的資料庫和本地資料庫之間進行遷移

同構遷移示例



異構遷移示例

您可以使用 AWS Database Migration Service (AWS DMS) 將現有的資料庫遷移或複製到 Amazon RDS。AWS DMS 支援在最廣泛使用的資料庫之間遷移。

支持的來源資料庫包括 Oracle、Microsoft SQL Server、MySQL、MariaDB、PostgreSQL、IBM Db2 LUW、SAP、MongoDB 和 Amazon Aurora。目標資料庫引擎包括 Oracle、Microsoft SQL Server、PostgreSQL、MySQL、Amazon Redshift、SAP ASE、Amazon S3 和 Amazon DynamoDB。

AWS DMS 還支援同構（相同引擎）遷移和異構（不同引擎）遷移。

所示的第一個示例是同構轉換，將本地 Oracle 資料庫遷移到 Amazon RDS for Oracle Database。

第二個示例展示了異構遷移，將在 Amazon EC2 實例上運行的 MySQL 資料庫遷移到 Amazon Aurora。

AWS DMS 的主要功能



- 執行一次性遷移
- 或完成連續資料複製
 - 示例：配置從本地資料庫到 RDS 實例的連續資料複製
- **AWS Schema Conversion Tool (AWS SCT)** 支援在源和目標之間更改資料庫引擎
- 典型的遷移主要步驟：
 1. 創建目標資料庫
 2. 遷移資料庫 schema
 3. 設置資料複製過程
 4. 啟動數據傳輸，並確認完成
 5. 將生產切換到新資料庫（用於一次性遷移）



AWS DMS 可用於執行一次性遷移，也可用於完成兩個資料庫之間的連續資料複製。例如，您可以使用它來配置從本地資料庫到 RDS 實例的連續資料複製。

當您想要執行從一個資料庫引擎到另一個資料庫引擎的異構遷移時，可能需要使用 *AWS Schema Conversion Tool (AWS SCT)*。

典型的資料庫遷移涉及以下主要步驟：

1. 創建目標資料庫
2. 遷移資料庫 schema
3. 設置資料複製過程
4. 啟動數據傳輸，並確認完成
5. 將生產切換到新資料庫（用於一次性遷移）

使用 AWS Snowball Edge 和 AWS DMS



遷移資料不可行的情況：

- 數據庫太大
- 連接太慢
- 您有隱私和安全方面的考慮

使用 AWS Snowball Edge

- 無需使用互聯網即可實現數 TB 傳輸



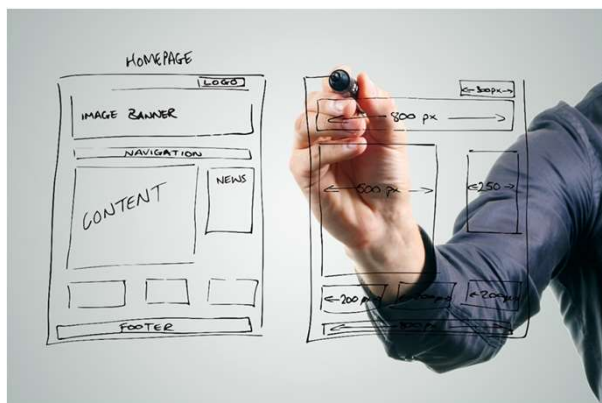
較大的資料移轉可能包含許多 TB 的資訊。由於網路頻寬的限制或資料量的限制，這一過程會很困難。AWS Database Migration Service (AWS DMS) 可以使用 [AWS Snowball Edge](#) 和 Amazon S3，比其他方法更快地遷移大型資料庫。

AWS Snowball 是一項 AWS 服務，此服務提供的 Edge 設備可讓您以比網路更快的速度將資料傳輸到雲。Edge 設備是 AWS 擁有的設備，它提供了大量的內置存儲空間。此外，Edge 設備使用 256 位元加密和開放標準的受信任平臺模組 (TPM)，可以確保您的資料安全並形成完整的監管鏈。

在您使用 Edge 設備時，資料移轉過程包括以下階段：

- 使用 AWS SCT 本地提取資料並將資料移至 Edge 設備
- 將設備寄回給 AWS
- 在 AWS 收到您的貨件後，該設備會自動將其資料載入到 S3 存儲桶中
- AWS DMS 提取相關檔並將資料移轉到目標資料存儲中

模組 5 – 挑戰實驗： 將資料庫遷移到 Amazon RDS

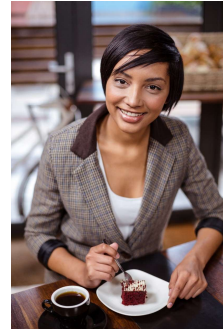


您現在將完成模塊 5 – 挑戰實驗：將資料庫遷移到 Amazon RDS。

業務需求：託管資料庫



對 Sofia 和 Nikhil 來說，在 EC2 實例上運行的資料庫變得難以維護。



最近 Olivia 訪問咖啡館時，向他們談到了 Amazon RDS 的功能。

Sofia 和 Nikhil 決定將咖啡館的資料庫遷移到 Amazon RDS。

對 Sofia 和 Nikhil 來說，咖啡館資料庫的管理變得很困難。例如，他們必須在咖啡館停止營業後加班加點進行每週的資料庫備份。

另外，他們最近正在努力安裝所需的補丁。Sofia 和 Nikhil 在週末下班後的時間幾乎沒有完成這項工作。Frank 和 Martha 現在意識到，這些維護工作增加了企業的人力成本，因為他們必須為 Sofia 和 Nikhil 的加班時間支付薪水。

AWS 解決方案架構師 Raquel 是咖啡館的顧客，也是 Sofia 的朋友。Sofia 在最近的一次談話中向 Olivia 提到了資料庫的主題。Olivia 建議他們將資料庫遷移到 Amazon RDS。

該解決方案將減輕手動執行常見資料庫維護任務的負擔，比如備份、補丁安裝和升級。作為一項完全託管的服務，Amazon RDS 會自動執行這些任務。

在本活動中，您將扮演 Nikhil 和 Sofia，負責將咖啡館資料庫遷移到 Amazon RDS。

挑戰實驗：任務

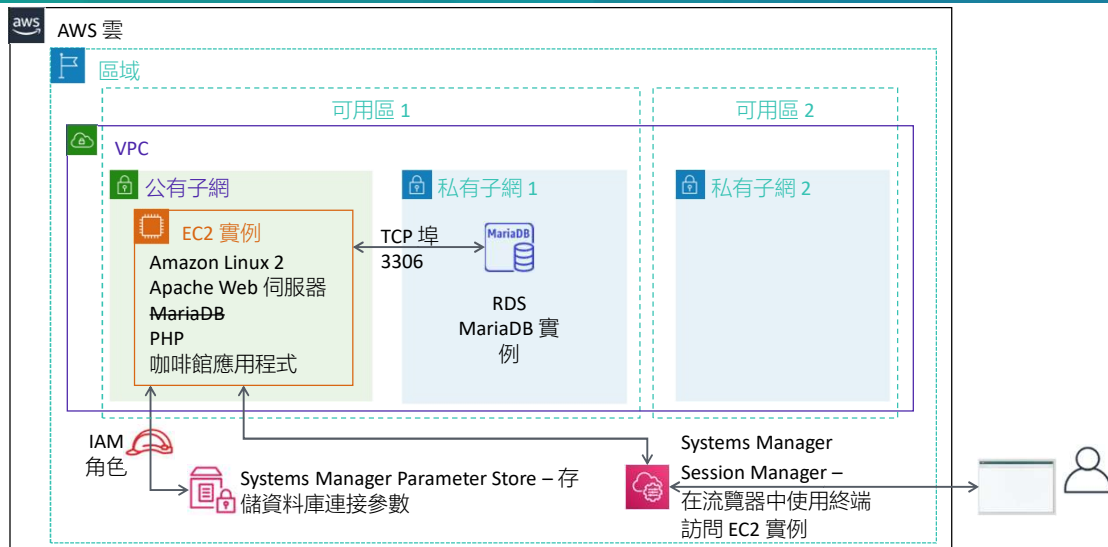


1. 創建 RDS 實例
2. 分析現有咖啡館應用程式部署
3. 使用 EC2 實例中的資料庫
4. 使用 RDS 資料庫
5. 將數據導入 RDS 資料庫實例
6. 將咖啡館應用程式連接到新資料庫

在本挑戰實驗中，您將完成以下任務：

1. 創建 RDS 實例
2. 分析現有咖啡館應用程式部署
3. 使用 EC2 實例中的資料庫
4. 使用 RDS 資料庫
5. 將數據導入 RDS 資料庫實例
6. 將咖啡館應用程式連接到新資料庫

挑戰實驗：最終產品



該圖總結了您完成實驗後將會構建的內容。

該咖啡館的 Web 應用程式原先使用的是安裝在 EC2 實例上的 MariaDB。在實驗過程中，您將創建一個 RDS MariaDB 實例，將 EC2 實例上的資料從資料庫中轉儲出來，並將這些資料移轉到 RDS 資料庫中。

為了完成遷移，您可以使用 AWS Systems Manager Session Manager（通過 Web 瀏覽器提供終端訪問）連接到 EC2 實例。在遷移過程中，您可以根據需要使用安裝在 EC2 實例上的 MySQL 用戶端連接到兩個資料庫。資料庫憑證和資料庫連接資訊存儲在 AWS Systems Manager Parameter Store 中。因此，在 EC2 實例中附加了一個 IAM 角色，以允許 Web 應用程式從參數倉庫中讀取資料。



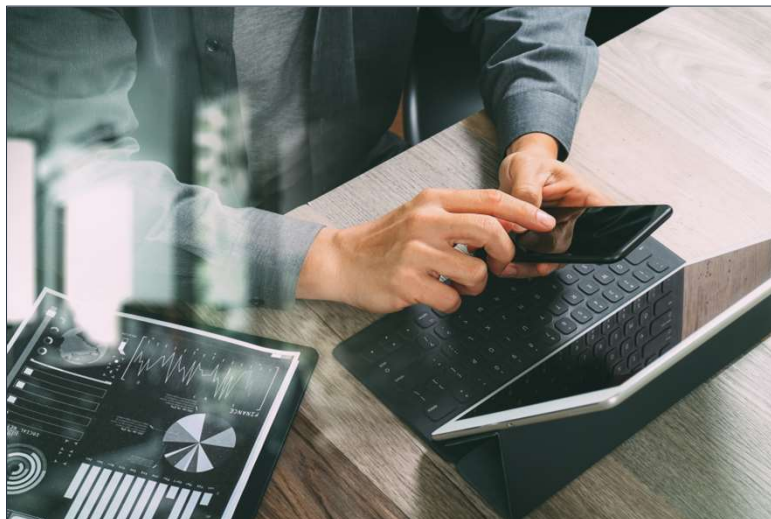
大約 80 分鐘



開始模組 5 – 挑戰實驗：將資料庫 遷移到 Amazon RDS

現在可以開始挑戰實驗了。

挑戰實驗總結： 要點



完成這個挑戰實驗之後，您的講師現在可能會帶您討論此挑戰實驗的要點。

模組 5：添加資料庫層

模組總結



現在來回顧下本模組，並對知識測驗和對實踐認證考試問題的討論進行總結。

模組總結



總體來說，您在本模組中學習了如何：

- 比較資料庫類型
- 區分託管服務和非託管服務
- 說明何時使用 Amazon Relational Database Service (Amazon RDS)
- 說明何時使用 Amazon DynamoDB
- 描述可用的資料庫安全控制
- 描述如何將資料移轉到 Amazon Web Services (AWS) 資料庫
- 部署資料庫伺服器

總體來說，您在本模組中學習了如何：

- 比較資料庫類型
- 區分託管服務和非託管服務
- 說明何時使用 Amazon Relational Database Service (Amazon RDS)
- 說明何時使用 Amazon DynamoDB
- 描述可用的資料庫安全控制
- 描述如何將資料移轉到 Amazon Web Services (AWS) 資料庫
- 部署資料庫伺服器

完成知識測驗



現在可以完成本模組的知識測驗。

一個應用程式需要一個高度可用的關聯式資料庫，初始存儲容量為 8TB。資料庫將每天增長 8GB。為了支援預期的流量，至少需要八個唯讀副本來處理資料庫讀取。

哪些選項將滿足這些要求？

- A. DynamoDB
- B. Amazon S3
- C. Amazon Aurora
- D. Amazon Redshift

請查看答案選項，並根據之前突出顯示的關鍵字排除錯誤選項。

正確答案是 C。 Amazon Aurora 是一個關聯式資料庫，可自動擴展以適應資料增長。Amazon Redshift 不支持唯讀副本，也不會自動擴展。DynamoDB 是 NoSQL 服務，而非關聯式資料庫。Amazon S3 是物件存儲，而非關聯式資料庫。

其他資源



- [AWS 資料庫 – 資源頁面](#)
- [Amazon RDS 入門指南](#)
- [Amazon RDS 最佳實踐](#)
- [Amazon RDS 常見問題](#)
- [Amazon DynamoDB 開發人員指南](#)
- [Amazon DynamoDB 常見問題](#)

如果您想瞭解有關本模組所涵蓋主題的更多資訊，下面這些其他資源可能會有所幫助：

- [AWS 資料庫 – 資源頁面](#)
- [Amazon RDS 入門指南](#)
- [Amazon RDS 最佳實踐](#)
- [Amazon RDS 常見問題](#)
- [Amazon DynamoDB 開發人員指南](#)
- [Amazon DynamoDB 常見問題](#)

謝謝

© 2020 Amazon Web Services, Inc. 或其附屬公司。保留所有權利。未經 Amazon Web Services, Inc. 事先書面許可，不得複製或轉載本文的部分或全部內容。禁止因商業目的複製、出借或出售本文。如有對本課程的糾正或回饋意見，請發送電子郵件至：aws-course-feedback@amazon.com。如有其他任何問題，請與我們聯繫：<https://aws.amazon.com/contact-us/aws-training/>。所有商標均為各自所有者的財產。



感謝您完成本模組的學習。