



分析的協同效應：整合 Azure Databricks 和 Microsoft Fabric



分析的協同效應：整合 Azure Databricks 和 Microsoft Fabric

3 /

利用 Azure Databricks 和 Microsoft Fabric
強化現代資料分析

4 /

利用 Azure Databricks 和 Microsoft Fabric
簡化分析工作負載

10 /

Azure Databricks 和 Microsoft Fabric 中的
獎牌架構

15 /

搭配 Azure Databricks 和 Microsoft Fabric
使用資料湖存放庫資料

22 /

搭配使用效果更佳：Azure
Databricks、Unity Catalog 和 Microsoft
Fabric Purview

26 /

Microsoft Fabric 中的 Data Factory 和
Azure Databricks 活動

28 /

利用生成式 AI 增強組織能力

33 /

探索實際使用案例與實際操作範例

42 /

利用 Azure Databricks 和 Microsoft Fabric
實現卓越

43 /

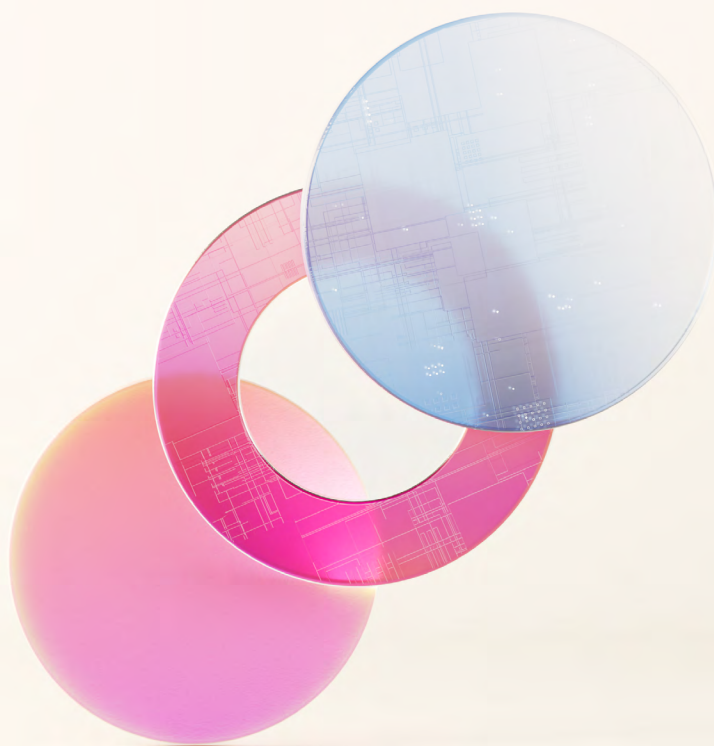
後續步驟

利用 Azure Databricks 和 Microsoft Fabric 強化現代資料分析

如今，組織需要了解如何以有效率且具洞察力的方式管理日益增加的大量資料。資料湖存放庫將資料湖的龐大儲存空間與各種資料服務的結構化處理相結合，支援廣泛的資料儲存和複雜的分析，完全不妥協。它不只是儲存解決方案而已，更增強了資料智慧並支援進階分析，應對將大量資料轉化為可行見解的挑戰。

雲端環境可隨需提供巨大的計算資源和可擴縮性；隨著組織資料的增長，其基礎結構可以無縫且符合成本效益的方式一同成長。雲端平台和資料湖存放庫架構之間的這種協同效應至關重要，為想要在資料驅動經濟中蓬勃發展的任何企業提供了具韌性和適應性的基礎。在雲端資料環境中，有效的管理和健全的安全性措施對於保護此一無價的資產或不可缺。

Azure Databricks 和 Microsoft Fabric 都是全方位的分析解決方案。Fabric 擁有較多方便商務使用者使用的工具，Azure Databricks 擁有整合式 AI 平台，但由於它們都依賴相同的資料層，因此可以做為更強大的整體一起使用。Azure Databricks、Fabric 和 OneLake 允許組織精簡其資料架構、簡化分析工作負載，以及跨統一平台實現有效率的資料管理和分析。



利用 Azure Databricks 和 Microsoft Fabric 簡化分析工作負載

現代資料湖存放庫架構可讓企業利用 Azure Databricks 與 Microsoft Fabric 之間的協同效應。Azure Databricks 和 Fabric 為廣泛的進階分析案例提供了一套統一、全面的工具，並合作提供一系列完整的解決方案，與資料湖存放庫搭配使用。它們具有資料工程、資料科學、資料倉儲和 Power BI 等元素，可提供廣泛的分析功能、為使用者提供具凝聚力的體驗，以及可供各種分析工具存取的單一資料存放庫。Azure Databricks 在單一的整合體驗中，也提供資料和 AI 資產的全面治理和譜系追蹤。

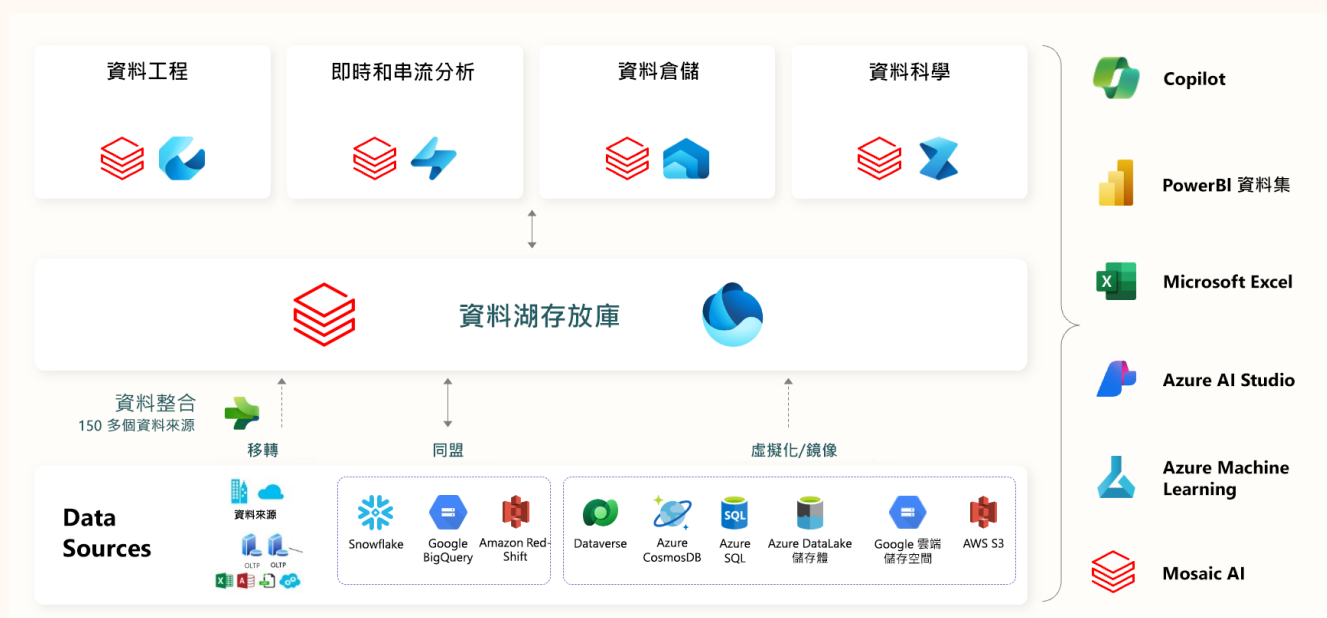


圖 1：資料湖存放庫架構中的 Azure Databricks 與 Microsoft Fabric 整合

利用 Azure Databricks 和 Microsoft Fabric 充分發揮資料潛力

Azure Databricks 與 Fabric 整合允許使用者在平台之間順暢切換，為使用者提供具凝聚力且強大的資料管理和分析解決方案，並輕鬆且有效率地促進 AI 和機器學習專案。

資料管理

OneLake 會集中來自不同來源的資料。Azure Databricks 與 Fabric 的整合不僅徹底改變了資料管理、可擴縮性和資料處理，更透過 OneLake 集中來自各種來源的資料。這種全方位的方法確保 Azure Databricks 可以無縫地連接儲存在 Azure Data Lake Storage (ADLS)、各種資料庫和 OneLake 本身當中的資料。如此可簡化對龐大資料量的管理，提高擴展資料專案的能力，並簡化資料處理管線。

- **集中式儲存**：在 Fabric 中使用 OneLake 允許集中式資料管理，這可簡化資料存取和治理，確保 Azure Databricks 直接將資料用於分析流程。
- **無縫整合**：Fabric 中 Azure Databricks 與 Data Factory 之間的順暢整合有助於簡化工作流程，包括資料擷取和驗證到轉換。此整合可實現具凝聚力的資料管理策略，支援資料分析、資料科學以及 AI 專案。
- **增強的安全性和可存取性**：進階 Azure Databricks 工作區支援憑證傳遞，加強了 OneLake 資源的安全性和存取便利性。此功能可確保以安全且直接的方式存取集中式資料，以進一步處理和分析。

可擴展性和可重現性

Azure Databricks 和 Fabric 支援可擴展的資料工作流程、可重現的 AI 和分析專案，以及動態資料處理功能。這種整合可讓組織有效率地管理大量資料，確保其資料環境中有一致的結果，並適應處理能力以滿足各種需求，推動可靠且可擴展的資料作業。

- **可擴展的資料工作流程**：Data Factory 中的 Azure Databricks 活動旨在支援可擴展的資料工作流程。組織可以有效率地處理大量資料，並根據需要擴展其資料處理和分析作業規模，而不會影響效能或可靠性。
- **可重現的 AI 專案**：整合確保 AI 和分析專案可重現，從資料版本控制及譜系追蹤等功能中受益。這些功能在這兩個平台上以原生方式提供，可提升 AI 專案的可靠性，並確保資料環境之間的一致性。

- **動態資料處理功能**：Azure Databricks 提供動態資料處理功能，可適應各種資料量和處理需求。這種彈性對於組織有效率地擴展其資料分析作業至關重要。

資料處理

這些技術結合了 Azure Databricks 健全的資料轉換功能與 Data Factory 管線複雜的協調流程，可發揮協同效應，以支援全方位的資料處理工作。

- **有效率的資料轉換**：Azure Databricks 擅長轉換儲存在 OneLake 和其他來源的資料。Azure Databricks 和 Fabric 共同支援廣泛的資料處理工作，包括資料探索、清理和準備，對於準備 AI 和機器學習的資料集至關重要。

- **協調複雜的工作流程**：包含 Azure Databricks 活動的 Data Factory 管線允許協調複雜的資料轉換工作流程。這些管線可以驗證資料來源、將資料複製到指定的儲存體，並執行筆記本進行資料轉換，為資料處理提供全方位的解決方案。

Azure Databricks 和 Fabric 之間的協同效應 (尤其是透過 Data Factory) 可改善資料管理、確保可擴縮性和可重現性，以及促進有效率的資料處理。這種整合對於組織使用其資料進行有洞察力的分析和 AI 驅動的決策至關重要。

資料湖存放庫架構的優點

資料湖存放庫架構是建基於開放原始碼 Delta Lake 儲存格式。除了其技術能力 (例如 ACID 交易一致性) 之外，它還提高了整個平台的整體效率。此外，它可讓多個處理引擎同時使用，因為它使用開放格式，並允許 Azure Databricks 和 Fabric 等工具同時處理的相同複本。

企業不必只依賴一個工具來處理其資料；相反地，他們可以為每個專案選取最佳工具。資料湖存放庫架構徹底改變了企業管理、擴展及處理資料方式。此創新方法透過交易支援確保資料完整性和一致性，以促進更有效的資料管理平台。此外，該架構將不同儲存格式在單一處融合的能力，簡化了資料資產的複雜環境。它有助於根據即時需求對運算資源進行動態調整，消除浪費的過度佈建，提高成本效益和資源利用。

此架構的核心是整合廣泛的資料湖和結構化資料倉儲，為推動 AI 和機器學習創新創造了最佳環境。這確保對計算能力與資料的存取，加速創新，以及簡化對各種資料系統的管理。

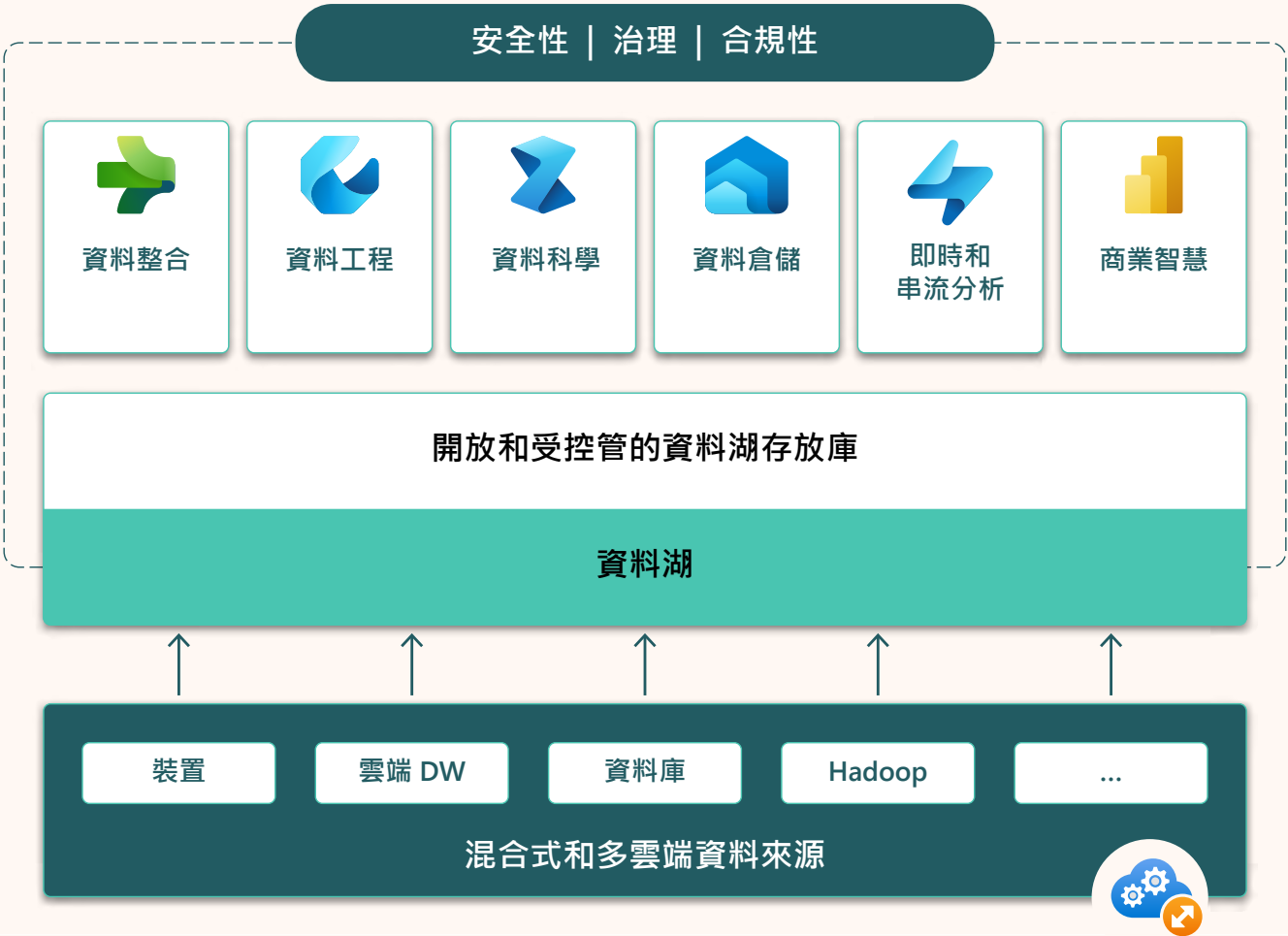


圖 2：資料湖存放庫的分析

資料湖的龐大儲存容量與資料倉儲的結構化、查詢最佳化環境的獨特組合，使現代資料湖存放庫成為開發及部署 AI 演算法的理想平台。這種雙重功能確保 AI 專案可以利用所需的計算能力和資料可存取性，加快創新並降低與管理不同資料系統相關的間接成本。

透過簡化資料架構並降低基礎結構複雜度，企業不必努力應對資料管理挑戰，而能透過以下方式專注於透過 AI 創造價值：

- 資料湖存放庫架構將 Delta Lake 檔案儲存在 ADLS 帳戶中。此雲端儲存服務極具成本效益，而 Delta Lake 格式允許儲存結構化和非結構化資料。
- 建立 AI 模型需要來自傳統 CPU 和進階 GPU 的大量運算能力。由於資料湖存放庫架構允許使用多個運算引擎（包括 Azure Databricks 和 Fabric），因此企業可以將適當的處理能力類型帶到其資料探索和資料模型化工作。

企業可以在資料湖存放庫內儲存的完整資料資產上使用 Azure Databricks 和 Fabric 的進階機器學習和 AI 功能。這些工具包括端對端實驗管理和自動化機器學習工具套件，可迅速推動 AI 專案。



Azure Databricks 和 Microsoft Fabric 中的獎牌架構

在更廣泛的資料湖存放庫架構概念中，獎牌架構是一種高度發展的方法，旨在簡化從擷取到見解的資料工作流程。其核心由三層所組成：銅級、銀級和金級，每一層在資料生命週期中都提供不同的用途。

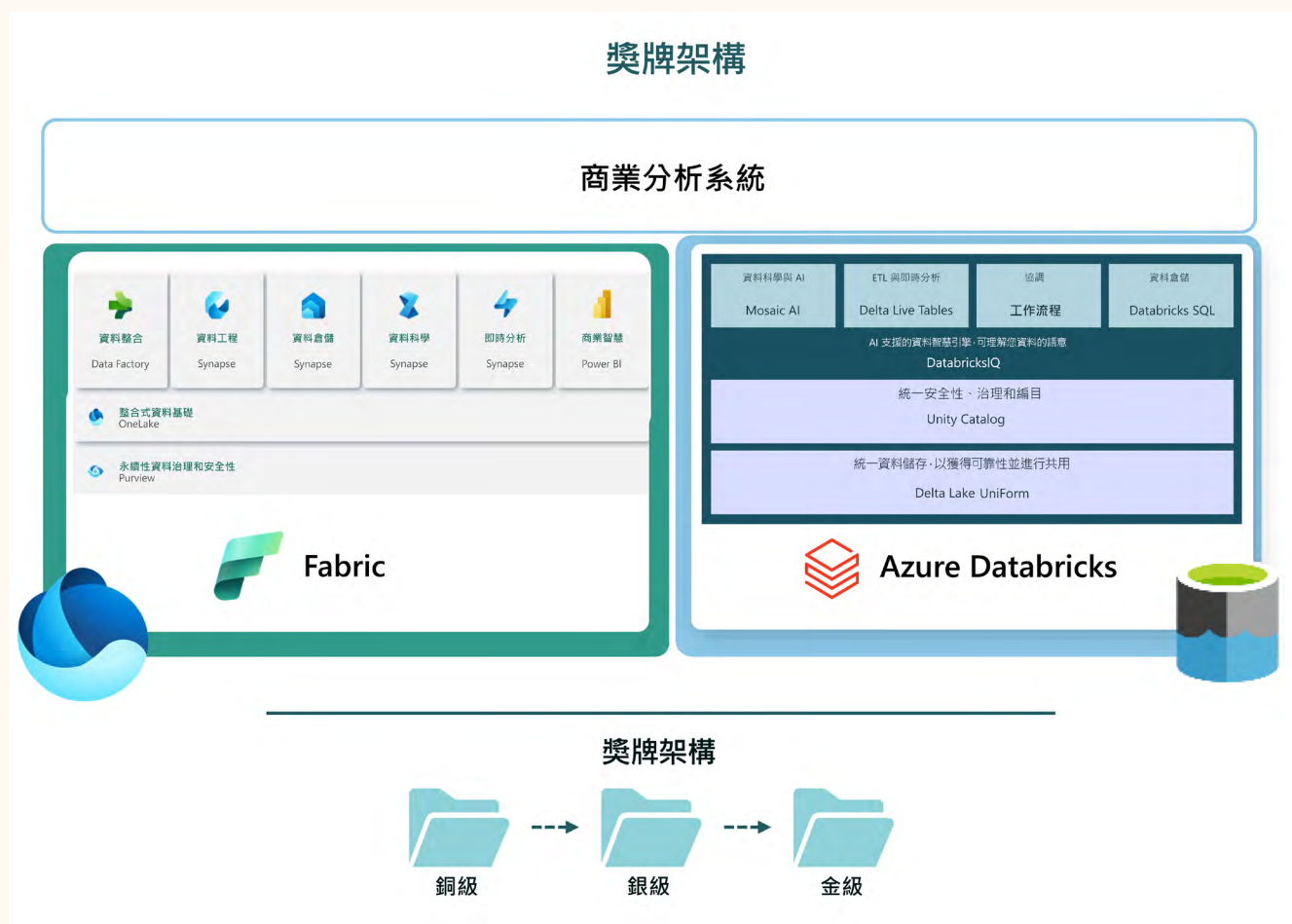


圖 3：獎牌架構

獎牌架構的三層是：

- 1. 銅級 (原始)：**在此層中，最初會擷取原始資料，保留其原始形式。它做為臨時區域，對於擷取完整細微度的資料而不遺失任何逼真度至關重要。
- 2. 銀級 (已驗證)：**在此層中，來自不同來源的資料會經過比對、整合並達成一致，為更複雜的分析工作做好準備。銀層旨在提供對關鍵業務實體的企業視野，對於支援自助式分析和中間資料儲存需求至關重要。
- 3. 金級 (豐富化)：**在此層中，資料已針對特定業務需求進一步最佳化，並且通常結構化為反正規化、針對讀取最佳化的格式，適用於高效能的查詢和報告。金層通常裝載直接用於商業智慧應用程式和影響支援系統的資料模型。資料在金層中成為真正的業務資產，提供寶貴且可操作的見解。

Azure Databricks 和 Fabric 利用此架構來增強其資料管理和分析方案。它們共同創造了一個健全的環境，其中資料不僅順暢地流過獎牌架構的每一個階段，而且更豐富且更容易存取。

與 Microsoft Fabric 和 Azure Databricks 的資料湖存放庫整合

資料湖存放庫優先模式代表了資料管理和分析的變革性方法。此方法建置在分層式資料儲存系統之上，將資料組織成銅層、銀層和金層。這種結構化資料流有助於更有效率的資料處理、分析和機器學習應用程式，將原始資料轉換為針對企業最佳化的資料。

Azure Databricks 利用其以 Spark 為基礎的分析引擎處理大量資料，有效地處理將資料從銅層轉換為銀層所需的資料轉換。

Fabric 提供了一個具凝聚力的分析平台，與 Azure Databricks 深度整合。它提供複雜的資料管理工具，並透過廣泛的連接器生態系統協助從各種來源順暢無縫地連接和擷取資料。

此整合可確保資料在獎牌架構的每一層自由移動，維持完整性和一致性，同時將複雜度和額外負荷減至最低。

開放原始碼儲存格式的基礎

採用 Apache Parquet 和 Delta Lake 使 OneLake 和 Azure Databricks 能夠將結構引擎最佳化，並加強跨平台的互通性。此策略可確保大型資料集獲得穩當的處理，跨資料湖存放庫架構促進順暢的資料存取，並降低通常與管理大規模資料架構相關的複雜度：

- **Apache Parquet 和 Delta Lake 的標準化**：OneLake 採用這些格式來處理大型資料集，並支援交易功能 (ACID 屬性)。這種標準化可確保整個 Fabric 引擎的所有資料都針對效能和相容性進行最佳化，從而促成更有效率的資料處理工作流程。

- **Apache Parquet 和 Delta Lake 的**

Fabric 引擎最佳化：透過重新設計資料處理引擎以針對這些格式最佳化，該系統可確保高效能的資料作業，而這對於有效率地處理大量資料至關重要。

- **全系統的互通性**：Azure Databricks 在 OneLake 中讀取任何 Fabric 成品的能力突顯了這些技術的互通性，確保資料可跨資料湖存放庫架構的不同部分無縫地進行存取和利用。

在資料湖存放庫架構中使用開放原始碼儲存格式可將資料利用率最大化、提高作業效率，並降低傳統上與管理大規模資料架構相關聯的複雜度。Apache Parquet 和 Delta Lake 對此有所幫助，確保資料以穩定且廣泛相容的格式儲存，讓組織能更輕鬆地跨不同的系統和平台整合及分析資料。

開放原始碼格式與獎牌架構整合

使用 Apache Parquet 和 Delta Lake 等開放原始碼格式，將此整合系統中的資料儲存和存取標準化，以支援進階資料管理功能，如 ACID 交易和結構描述演進。獎牌架構中 Azure Databricks 的處理能力與 Fabric 管理工具的強大組合，讓企業能以可擴展、高效且高度有益的方式著手資料架構，促進變革性見解的產生。

Azure Databricks 在資料處理和分析方面表現卓越，使用 Apache Spark 大規模執行健全的資料轉換和分析。它與 ADLS Gen2 整合，可讓 Azure Databricks 有效率地處理海量資料集，準備資料以進行進一步的分析處理。Fabric 透過提供其他資料管理工具來擴展 Azure Databricks 的功能，例如透過 200 多個原生連接器輕鬆存取資料來源，以及簡化的資料擷取機制。這可讓企業實施全方位的資料策略，涵蓋從擷取到洞察力分析的所有一切。

在此架構中，OneLake 協助集中化資料管理，而不需要實體移動資料。儲存在不同位置的資料可以像在單一存放庫中一樣進行存取和分析。Azure Databricks 的同盟功能對此進一步補充，允許跨不同資料存放區進行查詢，進而提高資料分析的彈性和範圍。

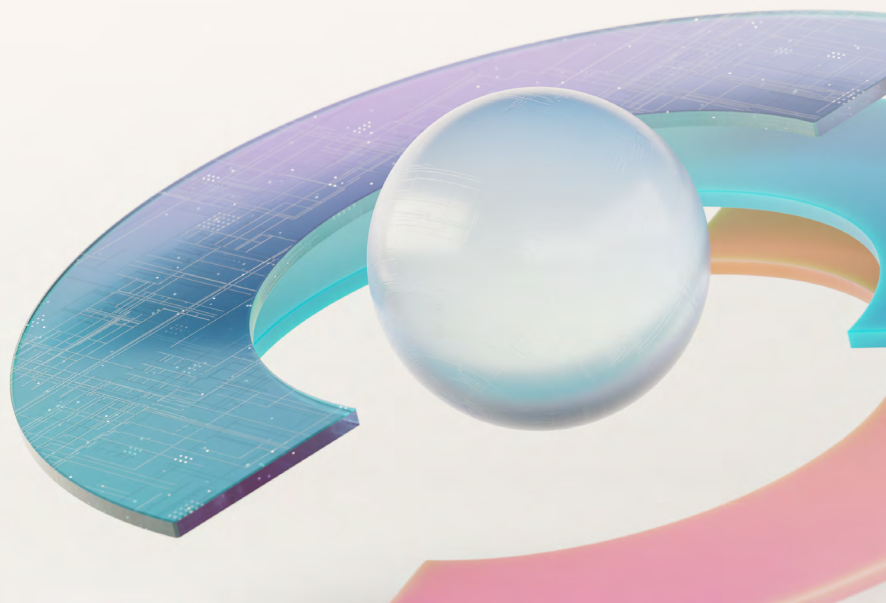
Azure Databricks 和 Fabric 之間的協同效應為建置進階資料湖存放庫架構提供了堅實的基礎。此組合可簡化不同資料來源的資料管理，並增進組織的分析功能，使它們更有效率且更準確獲得可操作的見解。這讓企業能夠以安全且可擴展的方式最大化其資料資產的價值。

資料湖存放庫優先模式案例

組織採用獎牌架構搭配 Azure Databricks 和 Fabric，將資料湖存放庫資料與 Fabric 工作負載整合，以及利用 OneLake 資料與資料湖存放庫同盟，讓組織可以在資料湖存放庫環境中建立複雜的分層儲存模型。這種統一方法促進了有效率的資料處理和管理。它支援對資料驅動的見解和操作需求的敏捷回應，同時提高跨多個儲存系統的資料分析的可擴縮性和彈性：

- **獎牌架構搭配 Azure Databricks 和 Fabric**：此方法在資料湖存放庫環境中利用分層儲存模型，促進有效率的資料處理和管理。組織可以搭配 Fabric 使用 Azure Databricks，更有效地管理大規模的資料分析管線。

- **將 Azure Databricks 資料湖存放庫資料與 Fabric 工作負載整合**：在此案例中，在 Azure Databricks 資料湖存放庫中儲存和管理的資料可直接與 Fabric 中的分析工具搭配使用。此整合支援更敏捷地回應資料驅動的見解和操作需求。
- **利用 Azure Databricks 中的 OneLake 資料與資料湖儲存庫同盟**：此設定允許跨 Azure Databricks 中的多個儲存系統來使用資料。使用者可以透過聯合資料來源跨這些來源查詢資料，就像它們是單一實體一樣，從而提高資料分析作業的彈性和可擴縮性。



搭配 Azure Databricks 和 Microsoft Fabric 使用資料湖存放庫資料

Azure Databricks 和 Microsoft Fabric 之間的協同效應為組織提供了一種強大且有效的方式來處理其資料工作負載。從擷取和儲存，乃至分析和報告，企業都能受益於安全且受治理的架構。這種彈性讓團隊能選擇最適合其專案需求的平台，確保在更廣泛的企業生態系統中的無縫整合。

例如，主要在筆記本編碼環境中工作的資料科學團隊將了解 Azure Databricks UI 的豐富功能，以及在叢集上管理進階 Spark 連結庫的靈活性，AI 工程師將感激對其資料微調模型的原生能力。商務分析師可能偏好在 Fabric 中輕鬆使用低程式碼資料流，以快速建置管線，在資料湖存放庫的金層中轉換資料並建立新資料集。兩個團隊都可以使用他們慣用的工具處理相同的資料集，而不需要任何一個團隊在自己環境中複製資料。

與資料湖存放庫資料互動

Azure 生態系統中的資料湖存放庫資料通常儲存在雲端位置，可以分為兩大類型：

- 1. ADLS 帳戶：**ADLS 是專為分析工作負載最佳化的雲端儲存系統。企業可以建立和管理 ADLS 帳戶，以滿足其資料管理需求。
- 2. OneLake：**OneLake 也是一種 ADLS 帳戶，但與其他帳戶不同，Azure 客戶不會直接管理它。相反地，它會做為 Fabric 的一部分並由 Fabric 管理。它不會顯示在 Azure 入口網站中，儘管客戶可以與其包含的資料互動，但他們無法控制該帳戶本身。

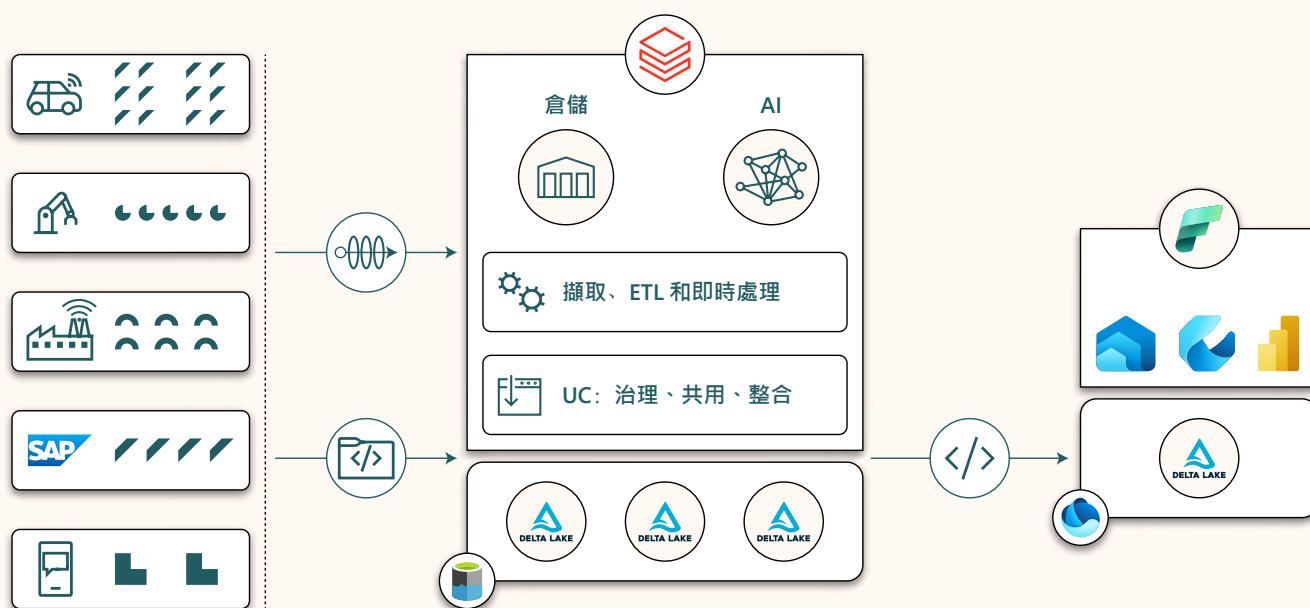


圖 4：使用捷徑將 ADLS 帳戶連結到 OneLake

OneLake 引入了捷徑，可協助資料專業人員跨 OneLake 和多個 ADLS 帳戶存取資料。捷徑允許資料專業人員將他們的 Unity Catalog (使用 Azure Databricks 捷徑) 或外部 ADLS 帳戶中的資料連結到 OneLake，使其看起來統一。使用者可以順暢無阻地存取這些帳戶的資料，而不需要知道它們來自不同的來源。捷徑透過將對外部資料來源的存取虛擬化並除去不必要的重複，支援可擴展且高效的 AI 模型訓練和部署流程，而有助於有效率地管理資料。

整合 Azure Databricks 與 Power BI 以加強資料視覺化

對於使用 Azure 資料湖存放庫的進階視覺化和儀表板案例，大多數企業選擇 Microsoft Power BI 做為其首選工具。這個強大的視覺化和分析工具現在已做為 Fabric 的一部分提供，允許企業將 Power BI 的管理和計費與其他 Fabric 資源完全整合。

Azure Databricks 與 Power BI 無縫整合。Databricks SQL 倉儲和 Unity Catalog 在資料湖存放庫中為 Power BI 提供了靈活且可擴展的解決方案。由 Azure Databricks 處理的資料有三種使用方式：

1. 針對 Power BI 的 Azure Databricks

直接發布：現在只需按個按鍵，Databricks 就可以自動將資料表（包括關聯）同步到 Power BI 語意模型。這可幫助分析師比以往更快的速度建立報告和儀表板。

2. Power BI Desktop 中的 Azure

Databricks 連接器：Azure Databricks 允許 Power BI 用戶端連接到 Azure Databricks 叢集，該叢集可以查詢及處理資料湖存放庫資料，並將結果傳送給 Power BI 以視覺化。

3. Power BI Direct Lake 模式：Power BI 可以使用其全新的 Direct Lake 模式直接讀取已寫入 Azure 儲存體位置的 Delta Lake 資料。這可以是 Azure Databricks 或 Fabric 所寫入的資料，而儲存位置可以是 OneLake 帳戶或任何其他 ADLS 帳戶。

上一節詳細介紹了如何使用 Azure Databricks 處理原始資料、準備報告，然後將資料寫入資料湖存放庫。

Power BI 中的 Direct Lake 模式用於讀取和視覺化 Azure Databricks 資料

使用 OneLake 儲存體，檔案會以有效的 Delta Lake 格式儲存。這些 Delta Lake 資料表已經過 VertiPaq 引擎最佳化，而能以高效的方式供 Power BI 取用。這可讓 Power BI 直接與儲存在 OneLake 中的 Delta Lake 資料表互動，而不需要中間的快取層，例如 Azure Analysis Services 或 Power BI 資料集。這種全新的存取模式稱為 Direct Lake 模式，提供即時資料存取，而不需要重新整理 Power BI 中的模型。

將資料集直接發佈到 Power BI 工作區：

- 從 Azure UI 發布，不使用 Power BI Desktop
- 使用資料表關聯 (PK/FK) 發布整個結構描述

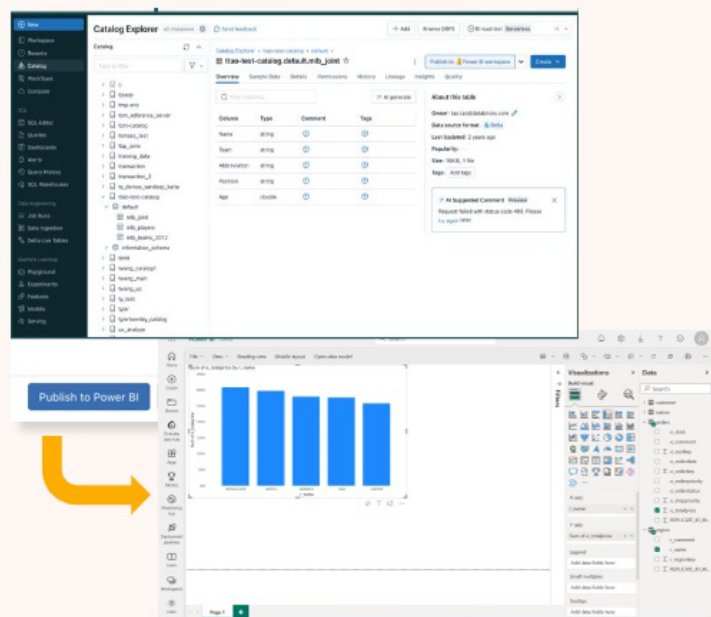


圖 5：深入 Power BI 整合

預設資料集包括資料湖存放庫的所有資料表，允許使用者建立關聯並套用各種模型變更。來自 Unity Catalog 的這些資料集可直接發佈到 Power BI。使用者可以使用可透過 Power BI 存取的 Web 模型化編輯器存取和編輯已發佈的語意模型。

在 Web 模型化編輯器中的模型檢視，您可以將滑鼠指標停留在資料表標題上，查看是否有 Direct Lake 連線。Direct Lake 還允許直接透過 Web 建立新的 Power BI 資料集。此程序可確保使用 Direct Lake 進行連線。若要深入了解將 Web 編輯器用於語意模型，下列文件將協助您開始著手：在 Power BI 服務 (預覽版) 中[編輯資料模型 - Power BI | Microsoft Learn](#)。

整合 Azure Databricks 與 Power BI 以加強資料工作流程

將 Azure Databricks 與 Power BI 整合可為資料管理和視覺化提供明顯的優勢，從而增強資料分析工作流程的安全性和績效：

1. 首先，整合允許直接從資料湖獲得更安全且互動式的資料視覺化體驗，避免與傳統資料處理工作流程相關的延遲和成本。它使用 Microsoft Entra ID 進行驗證，簡化使用者體驗並提高安全性，而不需要個人存取權杖。此整合可確保在 Power BI 中實作資料湖層級的安全性控制，在平台上維持一致的安全性策略。¹
2. 其次，語意資料湖存放庫架構可簡化資料的擷取和儲存。它提供統一的儲存層，支援廣泛的資料格式和結構，大幅提升資料處理和轉換的效率。此設定不僅簡化了分析堆疊，還提高了 BI 工具的資料品質和可存取性，而能直接對大型資料集進行更複雜的資料模型化和分析。²

3. 最後，整合支援進階分析案例，簡化對大型資料量的管理和分析。Power BI 中的 DirectQuery 選項在此扮演著關鍵的角色，允許使用者執行即時分析，而無需將資料從資料湖存放庫中移出。此功能對於維護報告和儀表板中的即時準確性至關重要，為企業提供深入且可立即行動的見解。³

這些功能共同使 Azure Databricks 和 Power BI 成為健全的組合，為企業提供了進階工具，以有效且安全地利用其資料。若要深入了解 Power BI 和 Azure lakehouse 整合，請參考以下文件：<https://learn.microsoft.com/fabric/get-started/directlake-overview>

³ <https://docs.databricks.com/partners/bi/power-bi.html>

¹ [在 Azure Databricks 中使用 Microsoft Power BI 和資料湖存放庫強化商業智慧：第 1 部分 - 基本知識 - Microsoft 社群中樞](#)

² [在 Azure Databricks 中使用 Power BI 和資料湖存放庫增強能力：第 3 部分 - 調整 Azure Databricks SQL - Microsoft 社群中樞](#)

使用 Data Activator 透過 Power BI 對 Azure Databricks 資料的變更發出警示

企業的一個常見案例是，他們希望當特定計量超過特定閾值時收到警示。例如，他們可能想知道特定品項的銷售量是否突然意外激增，或者他們可能想知道交易量是否大幅下降到低於正常範圍，指出交易管線中可能的問題。

這些案例可以由 Fabric 中稱為 Data Activator 的新功能來處理。此一無程式碼工具可監視 Power BI 報告中的資料，並在資料符合特定模式或達到指定的閾值時自動執行動作。發生這些事件時，Data Activator 可以執行諸如警示使用者或開始 Power Automate 工作流程等動作。

若要啟用 Data Activator，請按照此處的官方文件操作：<https://learn.microsoft.com/fabric/data-activator/>

若要使用 Data Activator 在 Power BI 報告中將冷凍櫃的溫度降到 30° F 以下時建立警示，請按照下列步驟在 Fabric 工作區中監視冷凍櫃中的溫度：

1. 確認您的 Power BI 報告 (包括冷凍櫃溫度資料) 已在線上發佈至具備 Premium 容量的 Fabric 工作區。
2. 選擇溫度視覺效果：
 - a. **存取報告**：開啟追蹤冷凍櫃溫度的特定 Power BI 報告。
 - b. **選取相關視覺效果**：尋找顯示冷凍櫃溫度的視覺效果。
3. 按一下溫度視覺效果右上角的省略符號 (...)，然後選取 **[設定警示]**，或使用 Power BI 工具列中找到的 **[設定警示]** 按鈕。
4. 在 **[設定警示]** 窗格中，指定要如何接收警示 (電子郵件或 Teams)。如果您的視覺效果包含多個冷凍櫃 (維度)，請使用 **[每個]** 下拉式清單來選取要監視的特定維度 (冷凍櫃)。
5. 定義警示條件，例如當低於 30° F 時。Data Activator 將監視溫度並在符合此條件時通知您。
6. 決定在 Power BI 中儲存 Data Activator 觸發程序的位置。您可以將它新增到現有的 Reflex 項目或建立新項目。

7. 按一下 **[建立警示]** 以底定您的 Data Activator 觸發程序。如果您希望在 Data Activator 中啟用觸發程序之前進行編輯，可以選擇取消選取 **[開始我的警示]**。

按照這些步驟操作，您已成功在 Data Activator 中設定警示，在受監視的冷凍櫃溫度低於 30° F 時通知您，讓您能夠在必要時立即行動。完成這些資料更新後，您應該會從設定的 Data Activator 收到警示。

使用資料湖存放庫監視與警示對 Azure Databricks 中的變更發出警示

企業通常需要在資料品質計量超過特定閾值時獲得警示。例如，他們可能想知道特定欄位中缺少值的數量是否突然意外激增，指出交易管線中可能有問題，或者機器學習模型的預測品質是否下降，表示需要針對較新的資料重新訓練模型。

這些案例可以使用 Azure Databricks 功能進行處理，稱為資料湖存放庫監視與警示。此一無程式碼工具可監視 Unity Catalog 中的資料品質，並在資料符合特定條件或超過閾值時自動執行動作。發生這些事件時，警示將執行指定的動作，例如透過電

子郵件、Slack 或 Teams 傳送通知。警示還可以呼叫 Webhook 動作，允許使用者根據資料中的變更建立可延伸的自訂工作流程。

監視器是按指定的排程執行的程序，以檢查特定資料表的資料品質。當使用者建立監視器時，它會計算資料表的資料品質計量，將目前的值儲存在不同的系統資料表中。每次監視器執行時，它會重新計算品質計量，並將其與原始值進行比較。如果品質退化，則引發警示。如需有關如何建立監視器的詳細資訊，請參考以下文件：

<https://docs.databricks.com/lakehouse-monitoring/create-monitor-ui.html>

如果監視器偵測到資料表中的資料品質下降，它將引發指定的警示。這可用於傳送通知給資料工程團隊，以便他們進一步調查。如需有關如何設定這些警示的詳細資訊，請查看以下文件：<https://docs.databricks.com/lakehouse-monitoring/monitor-alerts.html>

搭配使用效果更佳：Azure Databricks、Unity Catalog 和 Microsoft Fabric Purview

隨著分析需求增長和資料平台演變為更複雜的系統，治理平台（管理資料可用性、完整性和安全性）變得至關重要。在資料湖存放庫架構中，資料治理有助於確保資料已適當地編目、分類和管理。組織可以實施有效的資料治理來適當管理其資料，並使用它來推動業務價值。

資料湖存放庫架構中有效的資料治理需要實作管理資料的政策、程序和標準。這包括定義資料擁有權及管理、建立資料品質標準，以及實施資料安全性和合規性措施。為了提供這些關鍵的資料治理功能，Azure Databricks 和 Microsoft Fabric 提供了強大的現代功能。

Azure Databricks 中的 Unity Catalog

Azure Databricks 包含 Unity Catalog，它為組織的資料儲存位置提供集中的細微存取控制、資料存取的稽核，以及從擷取到所有資料工作負載的譜系追蹤，Azure Databricks 則提供資料行層級和資料列層級的存取控制和資料探索工具。它現在還包括系統資料表，這提供一種直接明瞭的方式來查詢稽核資料、計費資料和譜系。此外，AI 功能也支援 Unity Catalog 自動記載資料表和資料行、促進語意搜尋，以及協助顯示相關的資料產品。

Microsoft Purview 用於治理 Microsoft Fabric

Fabric 與 Microsoft Purview 整合進行資料治理、資訊保護和資料外洩防護。資訊保護功能可讓企業探索、分類和保護儲存在資料湖存放庫中的資料，並對其套用敏感度標籤。資料外洩防護使用原則來偵測敏感資料何時上傳到 Power BI 語意模型或其他支援的 Fabric 資源。它還可以協助偵測常見的敏感資料。Fabric 也包括用於探索資料譜系的工具，以便在資料從原始來源、移至各種報告模型時，能透過分析過程進行追蹤。

Microsoft Purview 和 Unity Catalog 用於簡化資料治理

Microsoft Purview 和 Unity Catalog 是兩種強大的工具，旨在提升雲端環境中的資料治理和管理，尤其適用於 Microsoft 提供之廣泛雲端服務的使用者。

Microsoft Purview 的廣大治理功能可以擴充到 Azure Databricks 環境中，其中的 Unity Catalog 會套用特定治理和安全性措施到 Azure Databricks 工作區。這種整合可讓組織在所有平台上維持一致的治理策略，從而提高安全性和操作效率。組織可以確保所有資料原則得到統一套用、資料譜系清楚且可稽核，並且整個資料區符合所有法規遵循要求。

Unity Catalog 在 Azure Databricks 資料湖存放庫平台中提供了高度發展的集中式治理解決方案，用於管理各種資料資產。它與 Azure 無縫整合，提供精細的治理功能，包括存取控制、稽核和資料譜系。Unity Catalog 可簡化跨多個 Azure Databricks 工作區的資料管理，讓組織可以跨其資料資產強制一致的安全性和合規性原則，無論是檔案、資料表還是機器學習模型。

Unity Catalog 為資料存取原則提供單一控制點，可跨所有工作區統一套用。這確保資料治理不僅集中化，而且深入整合到 Azure Databricks 環境中，從而增強安全性和治理。此外，Unity Catalog 還支援全面的資料探索，方便使用者尋找和存取他們所需的資料，同時遵守定義的存取控制和原則。這種統一方法有助於簡化作業，並降低通常與管理大型和多元化資料環境相關的複雜度。

Microsoft Purview 可讓企業透過 Fabric 維持對資料的控制權，實現從各種來源到詳細報告資料的無縫整合和管理。除了一套保護不同環境中敏感資料的工具外，Microsoft Purview 也使用敏感度標籤提供敏感資料探索、分類和保護等功能。它還促進了專為複雜環境（如 Power BI 語意模型）量身打造的全面稽核和資料外洩防護策略。

最佳做法

Azure Databricks 與 Microsoft Purview 的整合著重於在 Azure Databricks 環境中將資料治理和安全性最大化。此整合的主要最佳做法包括：

- **安全存取關鍵資料：**Microsoft Purview 可用來自動探索 Azure Databricks 中的資料並進行分類、視覺化資料譜系，以及有效管理存取控制。這可確保只有核准的人員才能存取敏感資料或關鍵資料，並確保所有資料原則在 Azure 服務間以一致方式套用。
- **使用兩個不同的連接器來管理中繼資料：**Microsoft Purview 為 Azure Databricks 提供兩個不同的連接器。大多數企業將使用 Azure Databricks Unity Catalog 連接器，因為 Unity Catalog 支援 Azure Databricks 中的許多現代功能。然而，對於尚未移轉至 Unity Catalog 且仍使用 Hive 管理其中繼資料的客戶，Microsoft Purview 也有一個 Azure Databricks Hive Metastore 連接器可供使用。

- **利用自訂規則集：**企業可以使用 Microsoft Purview 來掃描目錄、結構描述、資料表和檢視。做為最佳做法，除了使用由 Microsoft Purview 提供的規則集之外，企業應該使用自訂規則集。針對世界不同區域建立自訂規則集，透過只使用特定區域所需的分類規則，可以加快掃描過程。
- **使用標籤工具指出資料敏感度：**Microsoft Purview 中的標記工具可用在 Unity Catalog 資料上，以指出檔案和資料行的敏感度。這些標籤會隨著資料一起移動，並且可由 Microsoft 資料生態系統中的其他工具使用（如 SharePoint 和 Power BI）以自動套用資料處理原則。

結合 Microsoft Purview、OneLake 中的 Azure 安全性與 Unity Catalog 的功能，可支援具韌性和敏捷的資料治理策略，讓企業在數位環境中有效使用其資料資產。

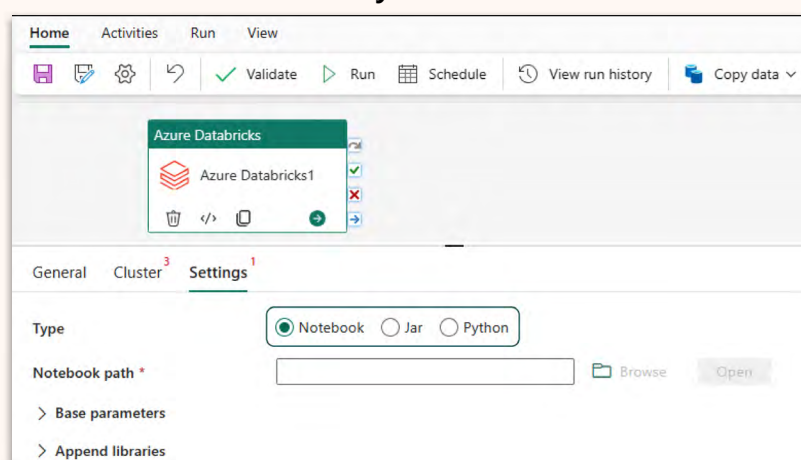


Microsoft Fabric 中的 Data Factory 和 Azure Databricks 活動

Microsoft Fabric 中的 Azure Databricks 活動代表了雲端環境中資料處理的重要演進，將 Azure Data Factory 廣泛的功能整合到一個更為統一且健全的架構中。使用新的 Azure Databricks 活動，使用者可以輕鬆地在 Fabric 中建立和管理資料管線，將複雜的分析和處理工作直接納入其工作流程中。使用者可以直接在 Fabric 中設定用於資料處理的 Azure Databricks 叢集，就像 Azure

Data Factory 中提供的功能一樣。這包括能夠設定 Azure 現成執行個體以更低成本存取未使用的 Azure 計算容量，並指定叢集原則以確保叢集設定符合組織標準和需求。

一項活動包含這三種工作類型：筆記本、Jar、Python



Unity Catalog 支援和原則 ID 整合

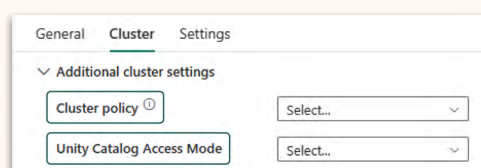


圖 6：Azure Databricks 和 Data Factory 與 Microsoft Fabric 之間的無縫整合

此外，經過改進的 Azure Databricks 整合引入了幾項新功能。包括：

- **設定 Unity Catalog 存取模式**：使用者可以設定 Unity Catalog 存取模式，以更周密的方式管理授權，從而提升治理和安全性。
- **在單一 Databricks 活動中執行多個任務**：使用者可以在單一 Databricks 活動中執行各種工作，例如筆記本、JAR 和 Python 指令碼，從而簡化流程並降低先前與管理多種類型的資料工作相關聯的複雜度。
- **即時監視筆記本執行**：在作業期間，使用者可以初始化資料管線並立即監視其執行情況，讓系統提供指向 Azure Databricks 執行個體中輸出的直接連結。此即時監視功能允許使用者直接從 Fabric 追蹤筆記本執行的詳細資訊，包括叢集效能和計算效率。

此整合不僅將 Azure Data Factory 現有的 Azure Databricks 功能引入 Fabric，還引入了如叢集原則和 Unity Catalog 支援等新功能，而提高了整體資料管理和分析體驗。



利用生成式 AI 增強組織能力

先進的 AI 模型已經改變了技術面貌。企業正在嘗試釋放其資料的潛力，並使用 AI 來擴展其商務能力。這包括使用生成式 AI 在 Azure 上建立其資料湖存放庫。

下一節將詳細介紹如何向 ChatGPT 描述所需的架構，並產生可在 Azure 中實作的程式碼。

利用 Azure OpenAI 產生架構和程式碼

使用提示工程，您可以使用用於摘要技術文章的相同流程來產生架構方案與程式碼。

提示輸入

我希望你擔任資料工程師。我會針對客戶需要利用 ADLS Gen2 中儲存的資料的情境提供詳細資訊。Azure Databricks 和 Microsoft Fabric 將存取它，以載入機器學習模型的資料。你的工作就是找出一個架構來滿足這個需求。

Azure OpenAI 的回答

為了處理客戶需要利用 Azure Data Lake Storage Gen2 (ADLS Gen2) 中儲存的資料的情境，並供 Azure Databricks 和 Microsoft Fabric 存取，以將資料載入機器學習模型，我們將設計一種有效率地利用這些服務的架構。此架構旨在簡化資料處理、加強資料治理，並確保可擴展的機器學習作業。

架構概觀

- 資料儲存層：Azure Data Lake Storage Gen2：做為中央存放庫，用於儲存原始資料、經處理的資料和機器學習成品。ADLS Gen2 已針對大規模分析案例進行最佳化，並支援階層命名空間，如此可簡化資料管理。
- 資料處理和分析層：Azure Databricks：做為資料處理、分析和機器學習模型開發的主要平台。Azure Databricks 提供了一個協作環境，支援多種程式設計語言，並與 ADLS Gen2 無縫整合，以直接存取資料。
- 資料管理和治理層：Microsoft Fabric：用於管理和治理整個資料環境的資料。Microsoft Fabric 允許建立統一的資料治理模型，當中包含 ADLS Gen2 中的資料以及 Azure Databricks 處理或產生的資料。
- 機器學習操作化：Azure Machine Learning：與 Azure Databricks 整合，用於部署和管理機器學習模型。此服務提供模型版本控制、監視和服務的工具，促進機器學習模型的操作化。

詳細工作流程

1. 資料擷取：資料從各種來源（包括結構化資料庫、IoT 裝置、記錄檔案等）擷取至 ADLS Gen2。此資料以原始格式儲存在階層式檔案系統結構中。

附註

Azure OpenAI 產生的其餘內容可在《附錄》中找到。

此處所反映的提示和答案可能無法反映您確切的體驗。

使用 Databricks Assistant 產生和更正程式碼

Databricks Assistant 是整合在 Azure Databricks 環境中的一種 AI 支援的工具，旨在協助使用者直接在 Databricks 筆記本中建立程式碼、解決錯誤和記載，以提高編碼工作效率。它使用 Azure AI 服務的功能，加強開發人員與平台中資料和程式碼的互動方式。

Databricks Assistant 在 Azure Databricks 工作區中充當健全的工具，可自動執行日常工作、最佳化程式碼、解釋功能和疑難排解，以支援開發人員。這不僅加快了開發過程，還有助於維持高標準的程式碼品質和文件，成為資料工程和分析工作流程中的資產。

Databricks Assistant 利用 AI 在 Azure Databricks 環境中以下列方式協助產生程式碼、解決錯誤和記載，來提高資料科學和工程的工作效率：

- **產生程式碼**：Databricks Assistant 允許使用者以自然語言輸入其需求，來簡化編碼工作。它能夠產生可執行的 SQL 查詢，或將程式碼從一種語言轉換為另一

種語言，例如將 Python Pandas 程式碼轉換為 PySpark。此功能可加快開發速度，並減少手動編碼錯誤。

- **解決錯誤**：Databricks Assistant 可以快速找出和釐清編碼錯誤，透過產生矯正性程式碼片段來提供解決方法。這對新手和有經驗的程式設計師都很有用，因為它可針對常見的語法和執行階段問題提供立即的解決方法，進而充分縮短停機時間。
- **記載**：它可協助記載程式碼，自動產生註解來解釋程式碼區塊的功能，支援維護清楚且可理解的程式碼基底，這對長期專案持續性和團隊協作至關重要。
- **情境式說明和學習**：Databricks Assistant 可協助掌握使用者的環境，包括經常使用的資料表、結構描述和先前查詢。它利用這個背景提供精確的答案和量身打造的程式碼片段，加強專案特定的支援。

- **視覺化和儀表板**：在像是 Lakeview 的視覺化環境中，Databricks Assistant 可以根據使用者提示來建立資料視覺效果，從而允許快速建立和反覆運算視覺顯示，而不需要深入的技術專業知識。
- **資料擷取和 ETL 程序**：Databricks Assistant 透過自動執行程式碼產生和最佳化、加快資料管線的設定和執行，來簡化資料擷取和 ETL 工作。
- **安全性與合規性**：Azure Databricks 確保與 Databricks Assistant 的互動是安全的，並遵循使用者權限和資料治理原則，使其適合在敏感和受監管的环境中使用。
- **整合和協助工具**：Databricks Assistant 可從筆記本、SQL 編輯器和檔案編輯器存取，成為適合各種資料工作的多功能工具。
- **回饋和反覆運算**：使用者可以直接透過平台提供回饋，如此有助於提高 Databricks Assistant 的準確性和功能。這可確保工具不斷演進，以因應使用者需求和挑戰。

請透過 14 天免費試用版佈建 Azure Databricks 工作區，當中包括對 Assistant 的存取權，讓潛在使用者可以評估其功能，並將其整合到工作流程中，完全無需初始投資。

以下是如何在不同案例中使用 Databricks Assistant 的一些範例，以提高工作效率和簡化工作：

1. 產生程式碼

案例：使用者需要從 DataFrame 按區域擷取和摘要銷售資料。

使用者輸入：

產生 SQL 查詢，從 sales_data 資料表按區域將所有銷售加總。

回應：

sqlCopy 程式碼

```
SELECT region, SUM(sales) AS  
total_sales FROM sales_data  
GROUP BY region;
```

效用：這可協助使用者快速取得所需的程式碼，而無需手動編寫 SQL 查詢。

2. 修正錯誤

案例：使用者撰寫了因語法錯誤而失敗的 PySpark 的程式碼。

使用者輸入：

此程式碼無法運作。
你可以幫忙修正嗎？

診斷和修正：Databricks Assistant 在 DataFrame 作業中找出一個遺漏的逗號，建議更正後的程式碼並醒目提示變更。

效用：使用者立即獲得回饋和更正建議，加快疑難排解並減輕挫折感。

3. 記載程式碼

案例：開發人員想要在複雜的函數中新增註解，以提高程式碼的可讀性。

使用者輸入：

你可以記載這個函數，解釋每個步驟嗎？

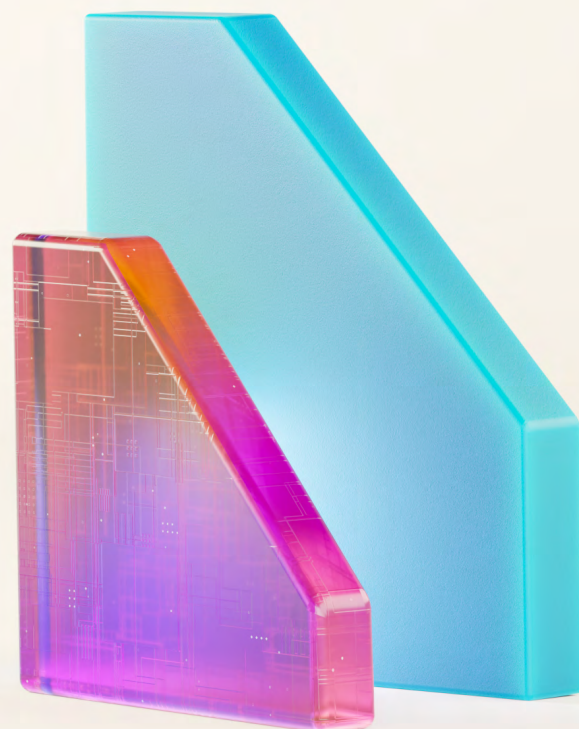
回應：Databricks Assistant 在每個重要行或程式碼區塊之前新增註解，解釋它執行的操作，例如初始化變數、錯誤處理和邏輯流。

效用：確保程式碼可理解，供未來參考或其他團隊成員使用，從而提高可維護性。

這些範例闡明了 Databricks Assistant 在真實世界開發環境中的實際優點、使編碼程序更具效率、簡化錯誤解析，以及確保周密的記載。

附註

此處所反映的提示和答案可能無法反映您確切的體驗。



探索實際使用案例與實際操作範例

在稍早的範例中，您使用 Python 程式碼讀取資料並進行彙總，以回答一些商務問題。本節將介紹用於讀取資料的 Python 程式碼的替代方案，以及如何使用 AI 來允許商務使用者使用英文而不是查詢語言來查詢資料湖存放庫資料。

使用適用於 Spark 的英文版 SDK 在 Azure Databricks 和 Fabric 中撰寫查詢

若要利用適用於 Apache Spark 的英文版 SDK，應滿足以下要求：

附註

Azure Databricks 建議使用 GPT-4。

- 1. 安裝英文版 SDK 套件：**一開始先將 SDK 新增到您的環境中。在筆記本中使用 `%pip install pyspark-ai --upgrade` 命令，以確保您擁有最新版本。
- 2. 重新啟動 Python 核心：**安裝後，您需要重新啟動 Python 核心以套用更新。在新儲存格中執行 `dbutils.library.restartPython()` 以重設環境。
- 3. 設定 OpenAI API 金鑰：**需要 OpenAI 的 API 金鑰進行驗證。實作方式為使用 `os.environ['OPENAI_API_KEY'] = '<your-openai-api-key>'` Python 程式碼設定環境，以實際的 API 金鑰取代 `<your-openai-api-key>`。
- 4. 啟用 SDK：**若要使用 SDK，請在筆記本中加以啟用。這包括使用慣用的語言模型（例如 GPT-4）來初始化 SDK，然後啟用它以開始解譯英文查詢。
- 5. 建立 DataFrame：**在筆記本中使用 SQL 查詢從 Azure Databricks 工作區擷取資料，並將其儲存為 DataFrame。此 DataFrame 將成為您進行英文查詢的基礎。

6. 使用英文查詢：最後，以簡明的英文提出問題來查詢 DataFrame。SDK 會解譯這些問題，並執行對應的 SQL 查詢，將結果直接傳回筆記本。

使用英文和適用於 Apache Spark 的英文版 SDK 進行查詢的範例可能如下：

2016 年 1 月每一天的平均行程距離為何？列出取至百分位的平均值。

此查詢示範如何使用簡明的英文來執行資料分析活動，例如使用英文版 SDK 從資料集計算平均值，而允許 Apache Spark 解譯和執行英文指示。

針對適用於 Apache Spark 的英文版 SDK 使用英文的另一個範例查詢可能是：

向我顯示上一季每個產品類別的總收入。

此類查詢說明使用者如何能夠使用自然語言，在定義的期間（例如上一季）請求特定財務計量（例如按類別細分的總收入）。此方法將複雜的資料分析工作簡化為直接了當的英文問題。

在 Microsoft Fabric 中建立筆記本

Fabric 筆記本是製作 Apache Spark 工作及進行機器學習實驗的關鍵工具。它支援進階視覺化和 Markdown 文字整合，提供一個 Web 互動式平台，深受編碼的資料科學家和工程師所歡迎。資料科學家依賴這些筆記本來開發和部署機器學習模型，包括實驗、模型追蹤和部署階段。Fabric 筆記本提供：

- 立即可用性，無需設定
- 用於資料探索和處理的直覺式低程式碼介面
- 透過整合的企業級功能的增強資料安全性
- 利用 Spark 的健全功能，分析各種格式的資料（包括 CSV、TXT、JSON、Parquet 和 Delta Lake）的能力

建立筆記本

建立筆記本時，使用者有兩個選項：建立新筆記本或匯入現有筆記本。組織可以遵循熟悉的 Fabric 項目建立工作流程來建立新筆記本：

1. 直接從 Fabric **[資料工程]** 或 **[資料科學]** 首頁，或整個工作區的 **[新增]** 選項初始化一個新筆記本。

2. 在同一視窗中選擇 **[匯入筆記本]**

以匯入現有的筆記本，例如 Azure Databricks 筆記本檔案。

3. 一旦開啟筆記本後，您可以新增程式碼，將資料寫入 OneLake。

在 OneLake 中處理資料非常簡單，並且並不涉及用於存取資料的複雜設定。

透過 Microsoft Fabric 資料工程筆記本將資料載入 OneLake

```
from pyspark.sql import SparkSession

# Initialize Spark session (assuming it's not already initialized)

spark = SparkSession.builder.appName("ParkDataImport").getOrCreate()

# URL to the CSV file

data_url = "https://www.dropbox.com/s/268uogek0mcypn9/park-data.csv?raw=1"

# Read the CSV data directly into a Spark DataFrame

df = spark.read.option("header", "true").csv(data_url)

# Assuming csv_table_name, parquet_table_name, and delta_table_name are
defined elsewhere in your code

csv_table_name = "park_data_csv"

parquet_table_name = "park_data_parquet"

delta_table_name = "park_data_delta"
```

```
# Save dataframe as CSV files to Files section of the default Lakehouse

df.write.mode("overwrite").format("csv").save("Files/" + csv_table_
name)

# Save dataframe as Parquet files to Files section of the default
Lakehouse

df.write.mode("overwrite").format("parquet").save("Files/" + parquet_
table_name)

# Save dataframe as a delta lake, parquet table to Tables section of
the default Lakehouse

df.write.mode("overwrite").format("delta").saveAsTable(delta_table_
name)

# Save the dataframe as a delta lake, appending the data to an existing
table

# Make sure the table exists and the schema matches to avoid errors

df.write.mode("append").format("delta").saveAsTable(delta_table_name)
```

讀取和資料分析

成功上傳資料後，請嘗試讀取和分析資料：

```
# Basic Data Analysis

# Count of animal sightings by type (excluding squirrels)

animal_sightings = spark.sql("""

SELECT Animal_Type, COUNT(*) as Total_Sightings

FROM park_data_view

WHERE Animal_Type != 'Squirrel'

GROUP BY Animal_Type

ORDER BY Total_Sightings DESC
```



```
""")

animal_sightings.show()

# Average temperature and most common weather conditions

avg_temp = spark.sql("""

SELECT AVG(Temperature) as Average_Temperature

FROM park_data_view

""")

avg_temp.show()

common_weather = spark.sql("""

SELECT Weather, COUNT(*) as Frequency

FROM park_data_view

GROUP BY Weather

ORDER BY Frequency DESC

LIMIT 5

""")

common_weather.show()

# Total count of squirrel sightings

squirrel_sightings = spark.sql("""

SELECT COUNT(*) as Total_Squirrel_Sightings

FROM park_data_view

WHERE Animal_Type = 'Squirrel'

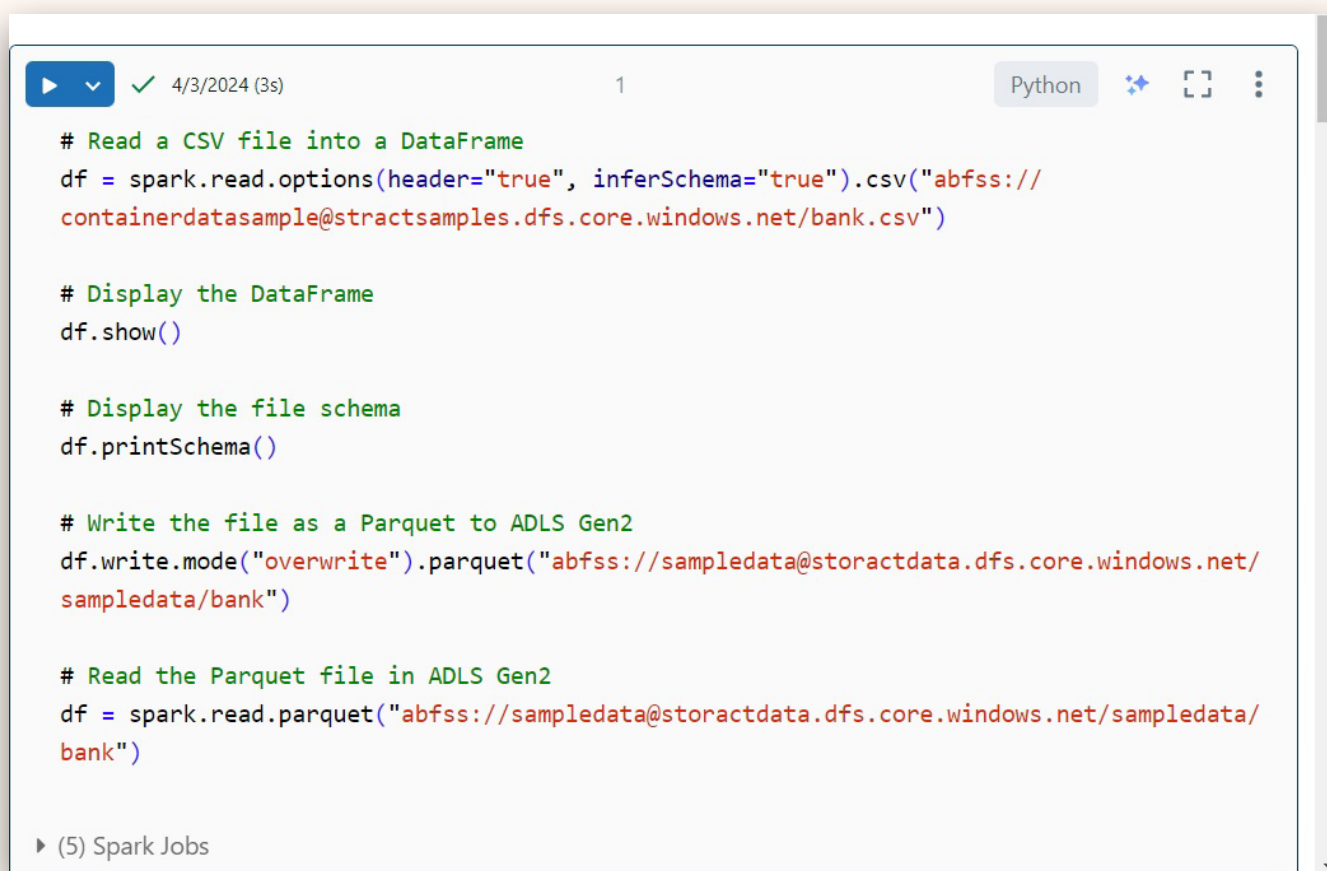
""")

squirrel_sightings.show()
```

在 Azure Databricks 中從 Parquet 建立和修改 Delta 資料表並將變更反映在 Fabric 中

Azure Databricks 和 Fabric 提供了一個資料湖存放庫環境，可讓企業使用不同的工具同時存取和分析其資料。這支援同一組資料上廣泛的資料處理活動，使組織能更輕鬆地有效率地管理和從其資訊中獲得見解。

1. 在所選的瀏覽器中開啟 Azure Databricks 工作區，並啟動新的 Azure Databricks 筆記本。



```
# Read a CSV file into a DataFrame
df = spark.read.options(header="true", inferSchema="true").csv("abfss://
containerdatasample@stractsamples.dfs.core.windows.net/bank.csv")

# Display the DataFrame
df.show()

# Display the file schema
df.printSchema()

# Write the file as a Parquet to ADLS Gen2
df.write.mode("overwrite").parquet("abfss://sampledata@storactdata.dfs.core.windows.net/
sampledata/bank")

# Read the Parquet file in ADLS Gen2
df = spark.read.parquet("abfss://sampledata@storactdata.dfs.core.windows.net/sampledata/
bank")
```

▶ (5) Spark Jobs

圖 7：範例筆記本

2. 將下列指令碼複製並貼到新的筆記本中。然後，在筆記本中執行下列 Python 指令碼，在 ADLS Gen2 帳戶中建立 Delta 資料表。此指令碼會讀取一些 Parquet 範例資料，然後將它作為 Delta 資料表寫入您的 ADLS 帳戶：

```
#python

# Adjust the file path to point to your sample parquet data using the
following format:

"abfss://<storage name>@<container name>.dfs.core.windows.
net/<filepath>"

# The line below reads Parquet files from your ADLS account

df = spark.read.format('Parquet').load("abfss://datasetsv1@olsdemo.dfs.
core.windows.net/demo/full/dimension_city/")

#This line writes the read data as Delta tables back into your ADLS
account

df.write.mode("overwrite").format("delta").save("abfss://
datasetsv1@olsdemo.dfs.core.windows.net/demo/adb_dim_city_delta/")
```

當然，Azure Databricks 也可以讀取 ADLS 帳戶中的資料。

3. Azure Databricks 還可以修改先前使用 Fabric 所建立的相同資料集合。若要了解其實際運作情況，附加一些新資料列到您在 OneLake 中建立的 Delta Lake 資料表：

```
# Import the necessary libraries

from pyspark.sql import SparkSession

from pyspark.sql.functions import lit

# Initialize a Spark session

spark = SparkSession.builder.appName("AppendToDeltaTable").
getOrCreate()
```

```
# Define the path to your Delta Lake table in OneLake

# Replace '<your-delta-table-path>' with the actual path to your Delta
Lake table

delta_table_path = "abfss://<container-name>@<storage-account-name>
dfs.core.windows.net/<your-delta-table-path>"

# Create a DataFrame with the new rows you want to append

# Replace the column names and values with those relevant to your table

new_rows = [

    ("NewValue1", 10),

    ("NewValue2", 20)

    # Add as many rows as needed

]

# Define the schema based on your Delta Lake table structure

# This is an example schema; adjust it to match your table's columns
and data types

schema = ["ColumnName1", "ColumnName2"]

# Create a DataFrame with the new data

new_data_df = spark.createDataFrame(new_rows, schema)

# Append the new data to the Delta Lake table

# Ensure the table format is set to 'delta' for Delta Lake
compatibility

new_data_df.write.format("delta").mode("append").save(delta_table_path)

# Verify by reading back the data from the Delta Lake table

df = spark.read.format("delta").load(delta_table_path)

df.show()
```

如範例所示，資料湖存放庫（建置的任何平台之上並具有開放式平台的優點），可讓企業使用各種引擎同時處理相同的資料複本。

Power BI 中的 Azure Databricks 連接器

適用於 Azure Databricks 的 Power BI 連接器可在 Power BI 和 Azure Databricks 之間順暢整合，讓組織輕鬆連接、分析和視覺化儲存在 Azure Databricks 中的資料。此整合支援 Microsoft Entra ID 驗證，而不需要管理員針對連線建立個人存取權杖。它旨在加強資料連線和分析體驗，而能直接從資料湖提供高效且安全的資料視覺化。

1. 取得 Azure Databricks 伺服器主機名稱和 HTTP 路徑，以在 Power BI 中設定連線。
2. 啟動 Power BI Desktop。
3. 從主畫面選擇 **[取得資料]**，或瀏覽至 **[檔案 > 取得資料]**。
4. 搜尋 **[Azure Databricks]**。
5. 選取 **[Azure Databricks 連接器]**，然後按一下 **[連接]**。

6. 輸入您稍早取得的伺服器主機名稱和 HTTP 路徑。
7. 針對資料連線決定 **[匯入]** 和 **[DirectQuery]** 模式。如需有關這些選項的詳細見解，請考慮閱讀有關 [Power BI Desktop 中的 DirectQuery](#) 使用方式。
8. 選擇您慣用的驗證方法：
 - a. **個人存取權杖**：輸入您的 Azure Databricks 個人存取權杖。
 - b. **Microsoft Entra ID**：選擇 **[登入]**，然後遵循提示操作。
 - c. **使用者名稱 / 密碼**：此選項通常不適用。
9. 驗證後，Power BI 將顯示 **[導覽]** 視窗。您可以在這裡選擇要查詢的 Azure Databricks 資料。如果工作區已啟用 Unity Catalog，請先選取目錄，然後選取結構描述和資料表。

對於需要 Azure Databricks 提供的處理能力和靈活性的工作負載，企業可以使用 Power BI 的進階視覺化功能與 Azure Databricks。

利用 Azure Databricks 和 Microsoft Fabric 實現卓越

Azure Databricks 與 Microsoft Fabric 的整合代表了在現代雲端環境中管理和分析資料的變革性方法。Azure Databricks 為資料處理和 AI 驅動的分析提供了一個高效能的平台，而 Fabric 則使用健全的資料管理工具來增強這些功能。此組合可讓組織更有效地利用進階分析和 AI 解決方案。

Azure Databricks 和 Fabric 透過提供順暢的資料存取，不需要冗餘的資料複本，而共同簡化了分析工作負載。此整合支援 Power BI 的直接查詢，從而提升效能並簡化資料架構。

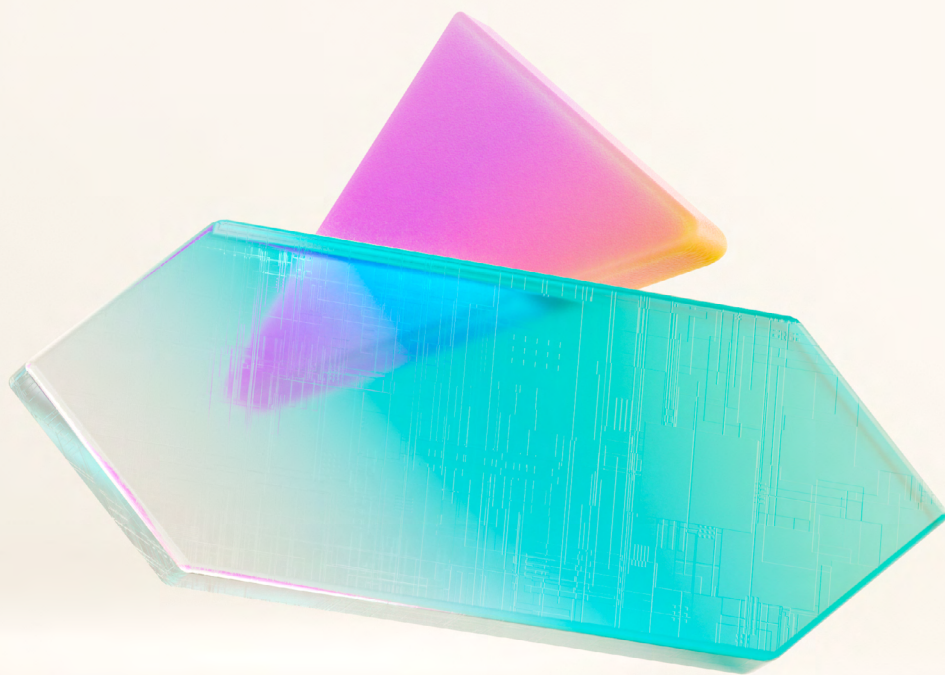
Azure Databricks 和 Fabric 之間的協同效應釋放了資料湖存放庫架構的全部潛力，加強了更有效地管理、處理和分析大型資料集的能力。這可讓組織輕鬆部署 AI 解決方案，並因應複雜的分析挑戰。

此外，Azure Databricks 和 Fabric 結合的功能可處理 AI 和機器學習專案的整個生命週期，從資料擷取和儲存，到分析和部署。此整合可協助組織維持對資料程序的控制，並確保 AI 計劃既可擴展且符合業界法規。

組織可以藉助 Azure Databricks 和 Fabric 的強大整合來提升其資料治理、提高營運效率，以及促進 AI 和分析創新，使它們在面對資料驅動經濟時擁有齊全的配備，並且幫助他們實現其策略性目標。

後續步驟

- 與銷售專家交談。
- 利用 Microsoft Fabric 從資料中獲取更多價值。
- 利用 Azure Databricks 從您的資料中發掘見解，並建立 AI 解決方案。
- 觀看網路研討會，深入了解 Azure Databricks 和 Microsoft Fabric 如何搭配使用發揮更佳效能。



附錄

提示輸入

我希望你擔任資料工程師。我會針對客戶需要利用 ADLS Gen2 中儲存的資料的情境提供詳細資訊。Azure Databricks 和 Microsoft Fabric 將存取它，以載入機器學習模型的資料。你的工作就是找出一個架構來滿足這個需求。

Azure OpenAI 的回答

為了處理客戶需要利用 Azure Data Lake Storage Gen2 (ADLS Gen2) 中儲存的資料的情境，並供 Azure Databricks 和 Microsoft Fabric 存取，以將資料載入機器學習模型，我們將設計一種有效率地利用這些服務的架構。此架構旨在簡化資料處理、加強資料治理，並確保可擴展的機器學習作業。

架構概觀

- 資料儲存層：Azure Data Lake Storage Gen2：做為中央存放庫，用於儲存原始資料、經處理的資料和機器學習成品。ADLS Gen2 已針對大規模分析案例進行最佳化，並支援階層命名空間，如此可簡化資料管理。
- 資料處理和分析層：Azure Databricks：做為資料處理、分析和機器學習模型開發的主要平台。Azure Databricks 提供了一個協作環境，支援多種程式設計語言，並與 ADLS Gen2 無縫整合，以直接存取資料。
- 資料管理和治理層：Microsoft Fabric：用於管理和治理整個資料環境的資料。Microsoft Fabric 允許建立統一的資料治理模型，當中包含 ADLS Gen2 中的資料以及 Azure Databricks 處理或產生的資料。
- 機器學習操作化：Azure Machine Learning：與 Azure Databricks 整合，用於部署和管理機器學習模型。此服務提供模型版本控制、監視和服務的工具，促進機器學習模型的操作化。

詳細工作流程

1. 資料擷取：資料從各種來源（包括結構化資料庫、IoT 裝置、記錄檔案等）擷取至 ADLS Gen2。此資料以原始格式儲存在階層式檔案系統結構中。
2. 資料處理和探索：
 - Azure Databricks 筆記本用於存取儲存在 ADLS Gen2 中的原始資料。資料工程師和科學家使用 Spark 處理和探索此資料，執行資料清理、轉換和特徵工程等任務。
 - 經處理的資料可以最佳化的格式（例如，Delta Lake）存回 ADLS Gen2，以進一步用於分析或機器學習。
3. 資料治理和譜系：
 - Microsoft Fabric 已經過設定，可對 ADLS-Gen2 中的資料資產以及 Azure Databricks 所產生的資料產品進行編目。它提供資料、強制原則、資料品質規則和安全性措施的統一視野。
 - Microsoft Fabric 可設定資料譜系和追蹤，在整個資料生命週期中保持資料轉換和使用量的可見度。
4. 機器學習模型訓練和部署：
 - 資料科學家使用 Azure Databricks 來開發及訓練機器學習模型，利用整合的 MLflow 進行實驗追蹤和模型管理。
 - 經過訓練的模型會在 Azure Machine Learning 中註冊，並使用 Azure Machine Learning 將這些模型以 Web 服務或容器的形式部署到正式環境中，進行即時或批次評分。

5. 監控和管理：

- Azure Machine Learning 提供模型監視功能，隨著時間的推移追蹤模型效能和資料漂移。它會在當模型過時時幫助重新訓練工作流程。
- Microsoft Fabric 會監督治理和合規層面，確保資料存取和模型部署遵守組織和法規標準。

此架構提供了一個全面的解決方案，可運用 Azure Databricks 和 Microsoft Fabric 中的 ADLS Gen2 資料，促進有效率的機器學習模型開發、治理和操作化。它確保可擴縮性、安全性和合規性，以滿足現代資料驅動組織的動態需求。

附註

此處所反映的提示和答案可能無法反映您確切的體驗。

提示輸入

基於上述，你可以提供詳細的建置指示並包含程式碼嗎？

Azure OpenAI 的回答

使用程式碼建立詳細的建置指南，將 Azure Data Lake Storage Gen2 (ADLS Gen2)、Azure Databricks 和 Microsoft Fabric 整合到具有凝聚力的機器學習架構中，需要幾個步驟。我們將在下面概述此程序的重要階段，包括設定、資料處理和機器學習模型生命週期管理。本指南假設對 Azure 服務和可在其中部署這些服務的訂閱已有基本的了解。

1. 設定和組態

Azure Data Lake Storage Gen2

1. 建立 ADLS Gen2 儲存體帳戶：

- 瀏覽至 Azure 入口網站。
- 建立新儲存體帳戶，選取您的訂閱、資源群組和「StorageV2」帳戶類型。
- 啟用階層命名空間以使用 ADLS Gen2 功能。

2. 設定儲存體帳戶存取：

- 在儲存體帳戶中，前往 [存取金鑰] 區段，記下連接字串供稍後使用。
- 或者，設定 Azure Active Directory (Microsoft Entra ID) 驗證，以更安全的方式存取。

Azure Databricks

1. 建立 Azure Databricks 工作區：

- 在 Azure 入口網站中，建立新的 Azure Databricks 服務。
- 部署完成後，瀏覽至 Azure Databricks 工作區並啟動工作區。

2. 在 Azure Databricks 中建立叢集：

- 在 Azure Databricks 工作區中，前往 [計算]。
- 建立新叢集，確保它與您計劃使用的程式庫相容。

3. 將 ADLS Gen2 裝載到 Azure Databricks (用於直接存取資料)：

- 在 Azure Databricks 筆記本中，使用以下程式碼裝載 ADLS Gen2 儲存體以直接存取：

Python

```
configs = { "fs.azure.account.auth.type": "OAuth", "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider", "fs.azure.account.oauth2.client.id": "<Application-ID>", "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="<scope-name>", key="<secret-key>"), "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/<Directory-ID>/oauth2/token" } dbutils.fs.mount( source="abfss://<filesystem>@<storage-account-name>.dfs.core.windows.net/", mount_point="/mnt/<mount-name>", extra_configs=configs)
```

- 以您實際的 Microsoft Entra ID 應用程式註冊詳細資訊和 ADLS Gen2 儲存體詳細資訊來取代預留位置。

Microsoft Fabric

1. 設定 Microsoft Fabric：

- 目前，Microsoft Fabric 是一個概念整合點，可能無法公開提供直接面向客戶的設定指示。通常，設定涉及設定您的環境，以跨 Azure Databricks 和 ADLS Gen2 進行編目和管理資料資產。
- 出於示範目的，我們將繼續執行下列步驟，假設 Microsoft Fabric 用於治理及保護資料存取，確保合規性和資料品質。

2. 使用 Azure Databricks 進行資料處理

1. 從 ADLS Gen2 載入資料：

- 使用裝載的路徑將資料讀入 Azure Databricks DataFrame：

pythonCopy 程式碼

```
df = spark.read.csv("/mnt/<mount-name>/path/to/data.csv")
```

2. 資料轉換：

- 使用 Spark DataFrame 作業執行必要的資料轉換。

3. 儲存經處理的資料：

- 以最佳化的格式（例如 Delta Lake）將經處理的資料存回 ADLS Gen2：

pythonCopy 程式碼

```
df.write.format("delta").save("/mnt/<mount-name>/path/to/  
processed/data/")
```

3. 機器學習模型訓練和部署

1. 模型訓練：

- 使用 Azure Databricks 使用經處理的資料來訓練機器學習模型。
- 使用 Spark MLlib 進行邏輯迴歸的範例：

pythonCopy 程式碼

```
from pyspark.ml.classification import LogisticRegression #  
Assuming df is a DataFrame prepared for training # Train a model  
lr = LogisticRegression(featuresCol='features', labelCol='label')  
model = lr.fit(df) # Save the model model.write().overwrite().  
save("/mnt/<mount-name>/path/to/model")
```

2. 模型部署：

- 與 Azure Machine Learning 整合，進行模型部署、監視和管理。此部分很複雜，通常涉及註冊模型、建立計分指令碼，以及將模型部署為 Web 服務。

附註

此處所反映的提示和答案可能無法反映您確切的體驗。

