Microsoft

分析中的协同效应：搭配使用 Azure Databricks 和 Microsoft Fabric

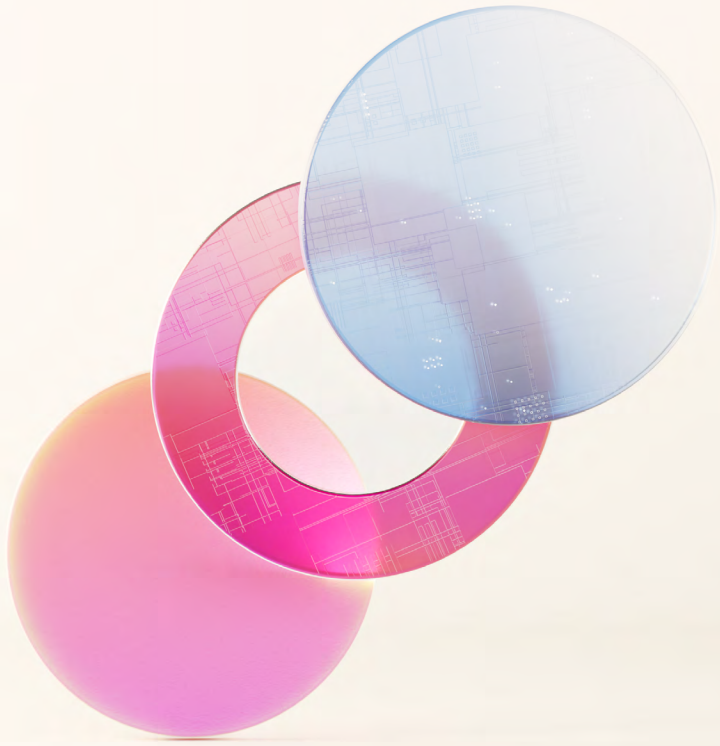
分析中的协同效应：搭配使用 Azure Databricks 和 Microsoft Fabric

Azure Databricks 与 Microsoft Fabric 助力现代数据分析

如今，组织需要了解如何以高效且富有见地的方式管理不断增长的海量数据。数据湖屋将数据湖的巨大存储能力与各种数据服务的结构化处理方式合二为一，能够支持广泛的数据存储和复杂的分析需求，游刃有余。它不仅仅是一种存储解决方案，还支持增强的数据智能和高级分析能力，能够将大量数据转换为切实可行的见解。

云环境可按需提供海量计算资源和扩展能力，其基础结构可以随着组织数据量的增长，以经济高效的方式无缝扩展。云平台和数据湖屋体系结构之间的这种协同作用至关重要，可以为任何希望在数据驱动型经济中蓬勃发展的企业提供复原能力和适应性基础。在基于云的数据环境中，有效的管理和强大的安全措施对于保护这一宝贵资产至关重要。

Azure Databricks 与 Microsoft Fabric 都是全面的分析解决方案。Fabric 具有更多业务用户友好型工具，Azure Databricks 具有一个集成式 AI 平台，但由于它们都依赖于同一数据层，因此可以作为一个整体使用，共同发挥更大的作用。借助 Azure Databricks、Fabric 和 OneLake，组织能够精简其数据体系结构，精简分析工作负载，在整个统一平台内进行高效数据管理和分析。



利用 Azure Databricks 和 Microsoft Fabric 简化分析工作负载

现代数据湖屋体系结构让企业能够利用 Azure Databricks 和 Microsoft Fabric 之间的协同效应。Azure Databricks 和 Fabric 各自提供了一套全面统一的工具，它们适合各种高级分析场景，还可以协同运行，为数据湖屋的使用提供完整的解决方案。它们集数据工程、数据科学、数据仓库和 Power BI 等各个领域的要素于一身，能够提供广泛的分析功能，为用户提供统一的体验，以及可用于各种分析工具的单个数据存储库。Azure Databricks 还能够以单独的统一体验，提供对数据和 AI 资产的全面治理和世系跟踪。

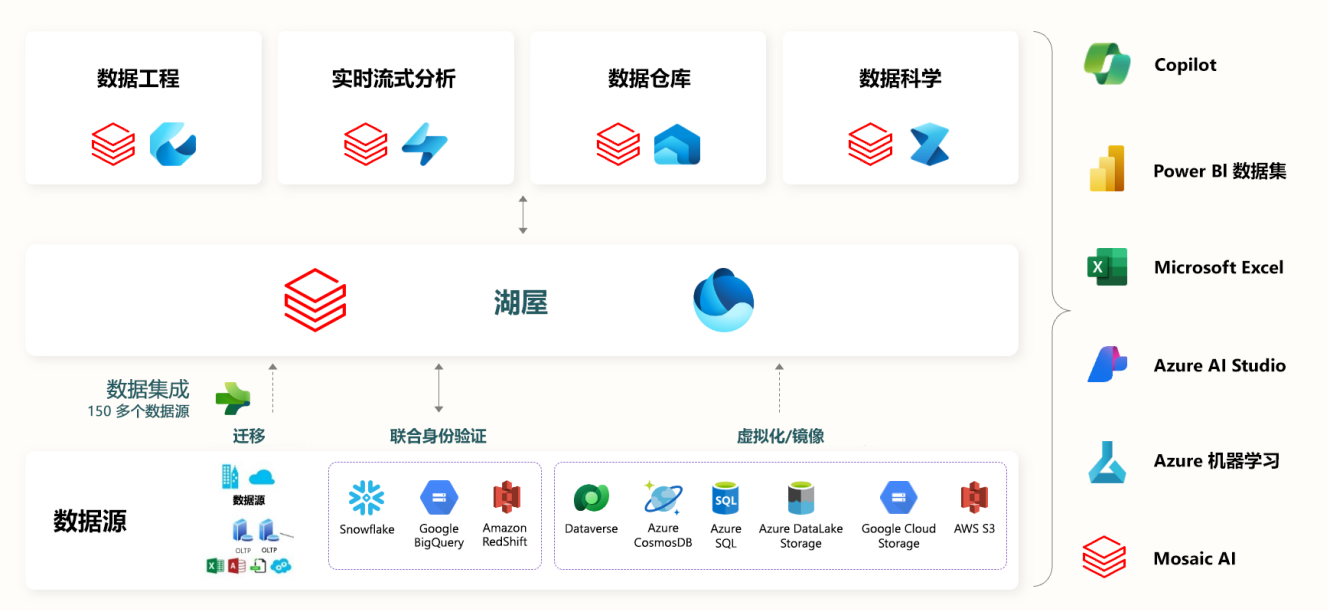


图 1：湖屋体系结构中的 Azure Databricks 与 Microsoft Fabric 集成

借助 Azure Databricks 与 Microsoft Fabric 最大限度发挥数据潜力

利用 Azure Databricks 与 Fabric 的集成，用户可以在平台间无缝切换，为客户提供统一且功能强大的数据管理和分析解决方案，还能轻松高效地推动 AI 和机器学习项目的开展。

数据管理

OneLake 可以将不同来源的数据集中在一起。Azure Databricks 与 Fabric 的集成不仅能够彻底改变数据管理、可扩展性和数据处理，还能通过 OneLake 将各种来源的数据集中在一起。这种全面的方法让 Azure Databricks 能够与 Azure Data Lake Storage (ADLS)、各种数据库和 OneLake 中存储的数据进行无缝连接。这可以精简大量数据卷的管理、增强扩展数据项目的能力并精简数据处理管道。

- **集中式存储**：在 Fabric 中使用 OneLake 可实现集中式数据管理，从而简化数据的访问和治理，确保 Azure Databricks 能够直接将数据用于分析过程。
- **无缝集成**：Azure Databricks 与 Fabric 内的数据工厂之间的无缝集成，有助于精简从数据引入和验证到数据转换的工作流。这种集成有助于实施统一的数据管理策略，为数据分析、数据科学和 AI 项目提供支持。
- **增强的安全性和可访问性**：高级 Azure Databricks 工作区支持凭据直通，可增强安全性并简化对 OneLake 资源的访问。利用这个功能，用户可以安全直接地访问集中式数据，进行进一步的处理和分析。

可扩展性和可再现性

Azure Databricks 与 Fabric 支持可扩展的数据工作流、可再现的 AI 和分析项目以及动态数据处理功能。这种集成让组织能够高效管理大量数据，确保在其数据环境中获得一致的结果，还能调整处理能力以满足各种要求，从而推动可靠且可扩展的数据操作。

- **可扩展的数据工作流**：数据工厂内的 Azure Databricks 活动旨在支持可扩展的数据工作流。组织可以高效处理大量数据，根据需要扩展其数据处理和分析操作，同时不对性能或可靠性造成影响。
- **可再现的 AI 项目**：该集成可确保 AI 和分析项目的可再现性，通过数据版本控制和世系跟踪等功能获益。这些功能在两个平台上本机可用，可增强 AI 项目的可靠性并确保各个数据环境间的一致性。
- **动态数据处理功能**：Azure Databricks 提供动态数据处理功能，可适应不同的数据量和处理要求。这种灵活性对组织有效地扩展其数据分析操作至关重要。

数据处理

Azure Databricks 强大的数据转换功能与数据工厂管道的复杂编排功能共同发挥作用，可支持各种数据处理任务。

- **高效的数据转换**：Azure Databricks 擅长转换 OneLake 和其他源中存储的数据。Azure Databricks 与 Fabric 共同支持广泛的数据处理任务，包括数据的探索、清理和准备，在 AI 和机器学习的数据集准备工作中发挥着重要的作用。

- **复杂工作流的编排**：包含 Azure Databricks 活动的数据工厂管道允许编排复杂的数据转换工作流。这些管道可验证数据源、将数据复制到指定存储，并执行笔记本以进行数据转换，为数据处理提供全面的解决方案。

Azure Databricks 与 Fabric 之间的协同效应，特别是通过数据工厂进行的协同工作，可增强数据管理、确保可扩展性和可再现性，并促进高效的数据处理。在组织使用数据进行富有见地的分析和 AI 驱动的决策方面，这种集成至关重要。

湖屋体系结构的优势

湖屋体系结构建立在开源 Delta Lake 存储格式的基础上。除了提供 ACID 事务一致性等技术功能外，它还能增强整个平台的整体效率。此外，由于它使用开放格式，允许 Azure Databricks 和 Fabric 等工具同时使用数据的同一副本，因此可以由多个处理引擎同时使用。

企业不必只依靠一种工具来处理数据；相反，他们可以为每个项目选择最合适的工具。湖屋体系结构彻底改变了企业管理、扩展和处理数据的方式。这种创新方法通过事务性支持确保数据的完整性和一致性，以打造更有效的数据管理平台。此外，该体系结构还能整合同一环境内的不同存储格式，简化数据资产的复杂格局。它可以根据实时需求动态调整计算资源、消除浪费的过度预配，提高成本效益和资源利用率。

这个体系结构的核心是广泛的数据湖与结构化数据仓库的集成，为促进 AI 和机器学习的创新营造一个最佳环境。这可以确保对计算能力和数据的访问，加快创新并简化各种数据系统的管理工作。

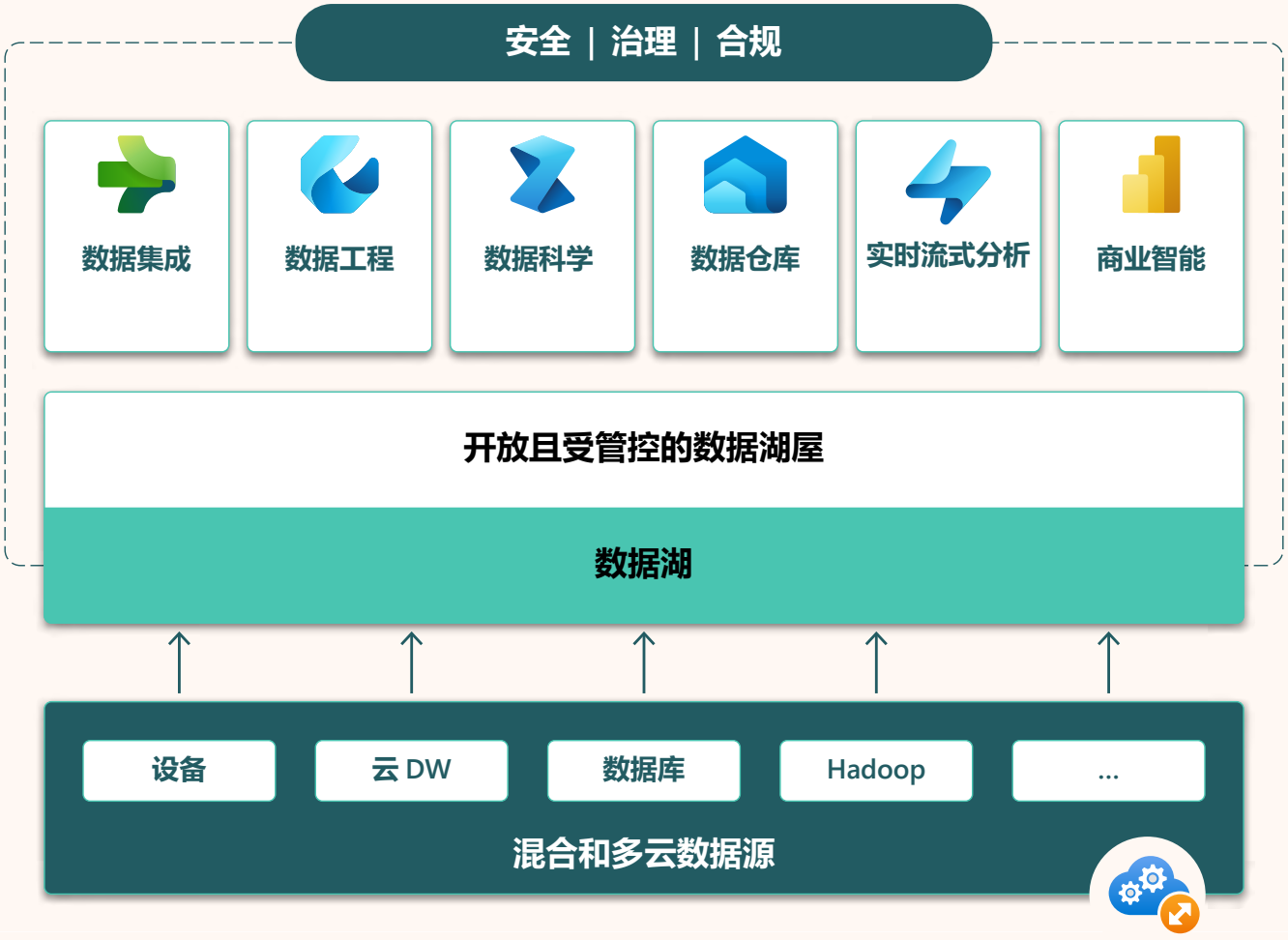


图 2：在湖屋内开展分析工作

现代湖屋将数据湖的巨大存储容量与数据仓库的结构化、查询优化型环境以独一无二的方式结合起来，打造开发和部署 AI 算法的理想平台。这种双重能力可确保 AI 项目能够利用必要的计算能力和数据可访问性，从而加快创新速度并降低与管理单独数据系统相关的开销成本。

通过以下方式，企业可简化数据体系结构并降低基础结构的复杂性，以便专注于通过 AI 创造价值，而不是忙于应对数据管理挑战：

- 湖屋体系结构将 Delta Lake 文件存储在 ADLS 帐户中。这种云存储服务极具成本效益，Delta Lake 格式允许同时存储结构化和非结构化数据。
- 构建 AI 模型需要来自传统 CPU 和高级 GPU 的大量计算能力。数据湖屋体系结构允许使用多个计算引擎，包括 Azure Databricks 和 Fabric，让企业能够提供适当的处理能力，以完成数据探索和数据建模任务。

企业可以将 Azure Databricks 和 Fabric 的高级机器学习和 AI 功能用于数据湖屋中存储的完整数据资产。其中包括端到端的实验管理和自动化机器学习工具包，可推动 AI 项目的进行。

Azure Databricks 与 Microsoft Fabric 中的勋章架构

在湖屋体系结构这个广泛的概念中，勋章架构是一种复杂的方法，用于精简从引入到生成见解的数据工作流。它的核心由三层组成：铜章层、银章层和金章层，每一层在数据生命周期中都发挥着不同的作用。

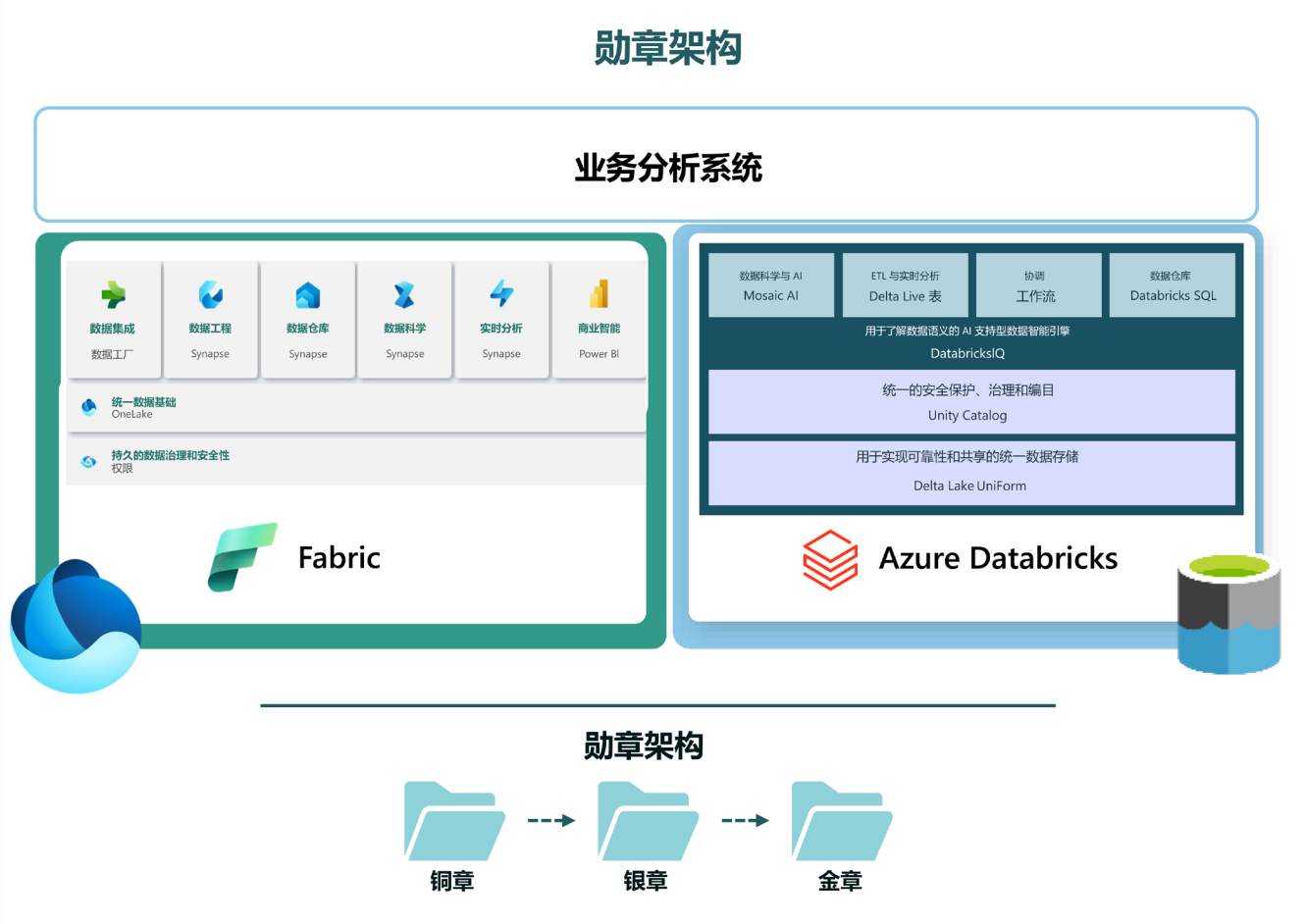


图 3：勋章架构

勋章架构的三层分别为：

1. **铜章层（原始）**：原始数据最初在这一层引入，并保留原始形式。它充当了一个暂存区域，对于在不损失保真度的情况下捕获全粒度数据至关重要。

2. **银章层（已验证）**：在这一层中，不同来源的数据将进行匹配、合并和一致化处理，为更复杂的分析任务做好准备。银章层旨在提供关键业务实体的企业级视图，对支持自助分析和中间数据存储需求至关重要。

3. **金章层（优化）**：在这一层中，数据将根据特定的业务需求获得进一步的优化，通常会被结构化为非规范化、读取优化型格式，适用于高性能查询和报告。金章层通常托管直接在商业智能应用程序和决策支持系统中使用的数据模型。在这一层，数据成为真正的业务资产，提供有价值、切实可行的见解。

Azure Databricks 与 Fabric 利用此架构来增强其数据管理和分析产品 / 服务。它们共同打造一个强大的环境，在这个环境中，数据不仅可以无缝流经勋章架构的每个阶段，还得到了扩充，变得更易于访问。

湖屋与 Azure Databricks 和 Microsoft Fabric 的集成

“湖屋优先”模式代表了一种数据管理和分析的变革性方法。这种方法基于分层式数据存储系统构建，将数据组织到铜章层、银章层和金章层中。这种结构化的数据流有助于提高数据处理、分析和机器学习应用程序的效率，将原始数据转换为已针对业务进行了优化的数据。

借助基于 Spark 的分析引擎，Azure Databricks 在处理大量数据方面颇有优势，可有效解决将数据从铜章层过渡到银章层所需的数据转换工作。

Fabric 提供了一个统一分析平台，可与 Azure Databricks 深度集成。它提供了复杂的数据管理工具，通过广泛的连接器生态系统，帮助从各种来源无缝连接和引入数据。

这种集成可确保数据能够自由通过勋章架构的每一层，在保持完整性和一致性的同时，最大限度地减少复杂性和开销。

开源存储格式的基础

通过采用 Apache Parquet 与 Delta Lake，OneLake 和 Azure Databricks 能够优化 Fabric 引擎并增强各平台间的互操作性。这个策略可确保面向大型数据集的强大处理能力，促进整个湖屋体系结构内的无缝数据访问，降低与管理大型数据体系结构相关的复杂性：

- **Apache Parquet 与 Delta Lake 的标准化**：OneLake 采用这些格式来处理大型数据集并提供事务性功能支持 (ACID 属性)。这种标准化可确保各个 Fabric 引擎上的所有数据都针对性能和兼容性进行了优化，从而实现更高效的数据处理工作流。
- **面向 Apache Parquet 和 Delta Lake 的 Fabric 引擎优化**：通过重新设计数据处理引擎，以针对这些格式进行优化，系统可确保高性能的数据操作，这对于高效处理大量数据至关重要。
- **整个系统内的互操作性**：Azure Databricks 能够读取 OneLake 中的任何 Fabric 工件，凸显了这些技术的互操作性，可确保数据在湖屋体系结构的不同部分之间得到无缝访问和利用。

在湖屋体系结构内使用开源存储格式可最大限度地提高数据利用率、改善运营效率，并降低大型数据体系结构管理方面始终存在的复杂性。通过确保数据以可靠且广泛兼容的格式存储，Apache Parquet 与 Delta Lake 做到了这一点，让组织能够更轻松地跨不同系统和平台来集成和分析数据。

开源格式与勋章架构的集成

Apache Parquet 和 Delta Lake 等开源格式的使用规范了这个集成系统内的数据存储和访问，支持高级数据管理功能，如 ACID 事务和架构演变。Azure Databricks 的处理功能与 Fabric 的管理工具在勋章架构内的强大组合，让企业能够以可扩展、高效且高度有益的方式处理数据体系结构，促进变革性见解的产生。

Azure Databricks 具有出色的数据处理和分析能力，可以使用 Apache Spark 大规模执行强大的数据转换和分析操作。 与 ADLS Gen2 的集成让 Azure Databricks 能够高效处理海量数据集，为进一步的分析处理做准备。Fabric 通过提供更多数据管理工具来扩展 Azure Databricks 的功能，例如通过 200 多个本机连接器和精简的数据引入机制轻松访问数据源。这让企业能够实施全面的数据策略，从数据引入一直到生成富有见地的分析结果，涵盖所有内容。

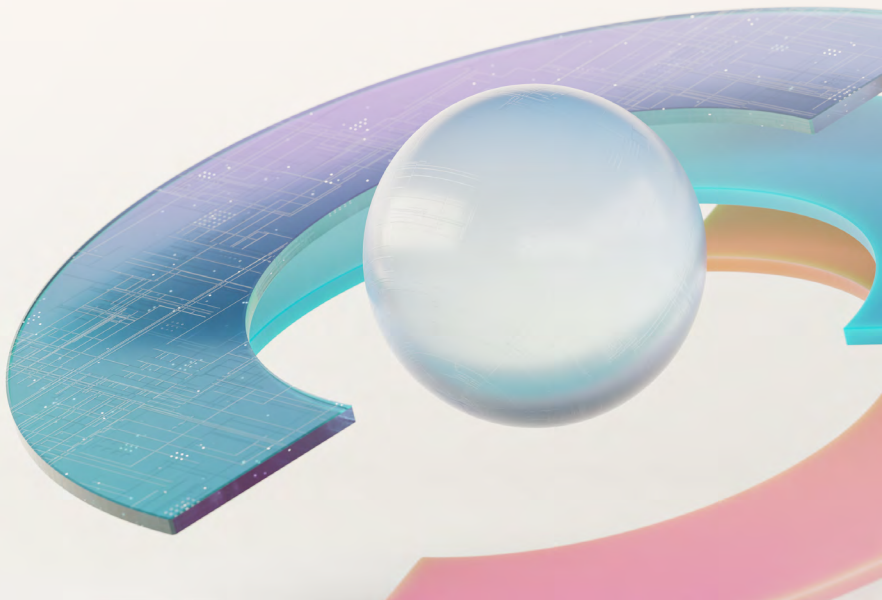
在这个体系结构中，OneLake 可帮助集中数据管理，无需进行物理数据移动。可以访问和分析存储在不同位置的数据，就像在单个存储库内一样。Azure Databricks 的联合功能允许跨不同数据存储区进行查询，进一步补充了这方面的能力，从而增强了数据分析的灵活性和范围。

Azure Databricks 与 Fabric 之间的协同作用为构建高级湖屋体系结构奠定了坚实的基础。这种组合可简化跨不同数据源的数据管理，增强组织的分析能力，让他们能够更高效、更准确地获得切实可行的见解，最终能够以可扩展的安全方式最大化数据资产的价值。

“湖屋优先”模式场景

通过采用带有 Azure Databricks 与 Fabric 的勋章架构，将湖屋数据与 Fabric 工作负载集成在一起，此外，利用 OneLake 数据与 Lakehouse Federation，组织可以在湖屋环境内实现复杂的分层存储模型。这种统一的方法有助于实现高效的数据处理和管理。它支持对数据驱动型见解和运营需求的敏捷响应，同时还能增强跨多个存储系统分析数据的可扩展性和灵活性：

- **包含 Azure Databricks 与 Fabric 的勋章架构**：这种方法利用了数据湖屋环境中的分层存储模型，能够促进高效的数据处理和管理。通过将 Azure Databricks 与 Fabric 结合使用，组织可以更有效地管理大规模数据分析管道。
- **将 Azure Databricks 湖屋数据与 Fabric 工作负载集成**：在这种情况下，在 Azure Databricks 湖屋中存储和管理的数据可以直接与 Fabric 中的分析工具一起使用。这种集成支持对数据驱动型见解和运营需求做出更敏捷的响应。
- **借助 Lakehouse Federation 在 Azure Databricks 中使用 OneLake 数据**：此设置允许在 Azure Databricks 内跨多个存储系统利用数据。通过将数据源联合起来，用户可以跨这些源查询数据，就像它们是单个实体一样，从而增强数据分析操作的灵活性和可扩展性。



通过 Azure Databricks 与 Microsoft Fabric 使用湖屋数据

Azure Databricks 与 Microsoft Fabric 之间的协同作用为组织提供了一种处理数据工作负载的强大有效方式。从引入、存储一直到分析和报告，企业都能通过一个安全且治理有序的框架获益。这种灵活性让团队能够选择最适合其项目需求的平台，确保在更广泛的企业生态系统内实现无缝集成。

例如，主要在笔记本编码环境中工作的数据科学团队会喜欢 Azure Databricks UI 的丰富功能以及在群集上管理高级 Spark 库的灵活性，而 AI 工程师会更愿意获得基于数据微调模型的原生能力。业务分析师可能更喜欢 Fabric 中易于使用的低代码数据流，帮助自己在湖屋的金牌层快速构建用于转换数据和创建新数据集的管道。所有团队都可以使用他们喜欢的工具来处理相同的数据集，无需将数据复制到自己的环境中。

与湖屋数据交互

Azure 生态系统中的湖屋数据通常存储在云位置，而这些位置可分为两种主要类型：

1. **ADLS 帐户**：ADLS 是一种云存储系统，面向分析工作负载进行了优化。企业可以创建和管理 ADLS 帐户，以满足其数据管理需求。
2. **OneLake**：OneLake 也是一种 ADLS 帐户，但与其他帐户不同，Azure 客户不直接管理这种帐户。相反，它是作为 Fabric 的一部分创建并由 Fabric 管理的。它不会显示在 Azure 门户中，尽管客户可以与其中包含的数据交互，但他们对帐户本身没有太多的控制权。

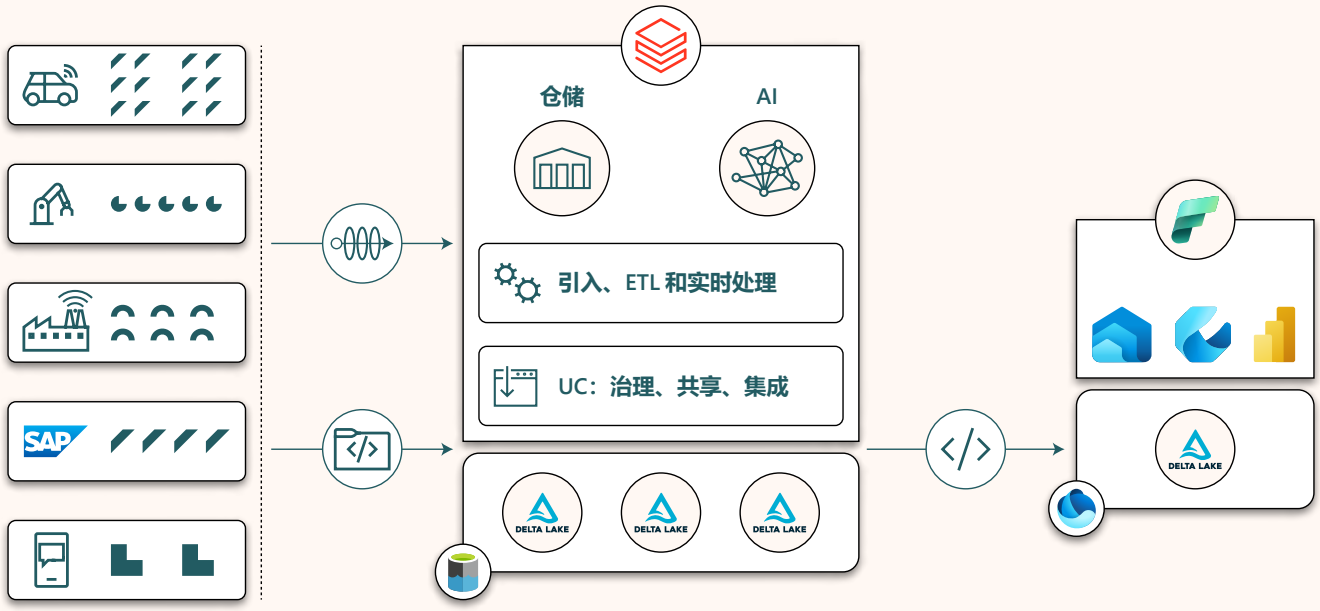


图 4：通过快捷方式将 ADLS 帐户关联到 OneLake

OneLake 引入了快捷方式，以帮助数据专业人员跨 OneLake 和多个 ADLS 帐户访问数据。借助快捷方式，数据专业人员可以将其 Unity Catalog（使用 Azure Databricks 快捷方式）或外部 ADLS 帐户中的数据关联到 OneLake，显示为一个整体。用户可以无缝访问这些帐户中的数据，不会意识到它们来自不同的来源。通过虚拟化对外部数据源的访问，减少不必要的重复过程，快捷方式可帮助有效管理数据，支持可扩展且高效的 AI 模型训练和部署流程。

将 Azure Databricks 与 Power BI 集成，以优化数据可视化效果

对于使用 Azure 数据湖屋的高级可视化和仪表板场景，大多数企业都选择 Microsoft Power BI 作为首选工具。这款强大的可视化和分析工具现在作为 Fabric 的组件提供，让企业可以将 Power BI 的管理和计费功能与其他 Fabric 资源完全集成起来。

Azure Databricks 可与 Power BI 无缝集成。Databricks SQL 仓库和 Unity Catalog 可以为湖屋中的 Power BI 提供灵活且可扩展的解决方案。由 Azure Databricks 处理过的数据可以通过以下三种方式使用：

1. **面向 Power BI 的 Azure Databricks 直接发布**：现在，只需单击一下，Databricks 就能自动将表（包括关系）同步到 Power BI 语义模型。这可以让分析师以前所未有的速度生成报告和仪表板。

2. **Power BI Desktop 中的 Azure Databricks 连接器**：Azure Databricks 允许 Power BI 客户端连接到 Azure Databricks 群集，后者可以查询和处理湖屋数据，然后将结果发送给 Power BI 以进行可视化呈现。

3. **Power BI Direct Lake 模式**：Power BI 可以使用全新的 Direct Lake 模式，直接读取已写入 Azure 存储位置的 Delta Lake 数据。这些数据可以是 Azure Databricks 写入的，也可以是 Fabric 写入的；存储位置可以是 OneLake 帐户，也可以是任何其他 ADLS 帐户。

上一节详细介绍了如何使用 Azure Databricks 处理原始数据、为报告做准备，然后将其写入湖屋。

Power BI 中用于读取和可视化 Azure Databricks 数据的 Direct Lake 模式

在 OneLake 存储中，文件以高效的 Delta Lake 格式存储。这些 Delta Lake 表已通过 VertiPaq 引擎进行了优化，能够高效地由 Power BI 使用。因此，Power BI 能够直接与 OneLake 中存储的 Delta Lake 表交互，无需借助 Azure Analysis Services 或 Power BI 数据集等中间缓存层。这种新的访问模式叫作 Direct Lake 模式，可提供实时数据访问，无需刷新 Power BI 中的模型。

直接将数据集发布到 Power BI 工作区：

- 通过 Azure UI 发布，无需 Power BI Desktop
- 发布具有表关系的整个架构 (PK/FK)

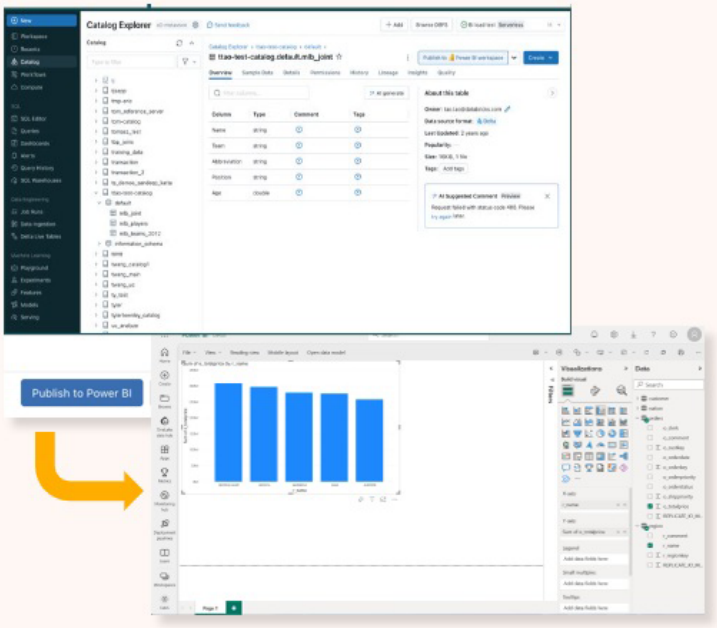


图 5：深度 Power BI 集成

默认数据集包括湖屋中的所有表，允许用户建立关系并应用各种建模更改。这些来自 Unity Catalog 的数据集可以直接发布到 Power BI。Power BI 提供了 Web 建模编辑器，用户可以通过该工具访问和编辑已发布的语义模型。

在 Web 建模编辑器的模型视图中，将光标悬停在表头上，即可查看是否存在 Direct Lake 连接。Direct Lake 还允许直接通过 Web 创建新的 Power BI 数据集。此过程可确保使用 Direct Lake 进行连接。要了解有关使用 Web 编辑器处理语义模型的详细信息，请参阅以下入门文档：**在 Power BI 服务中编辑数据模型（预览） - Power BI | Microsoft Learn**。

将 Azure Databricks 与 Power BI 集成以增强数据工作流

Azure Databricks 与 Power BI 的集成为数据管理和可视化提供了显著优势，增强了数据分析工作流的安全性和性能：

1. 首先，集成后，用户可直接通过数据湖获得更加安全的交互式数据可视化体验，避免了传统数据处理工作流造成的延迟和成本。它使用 Microsoft Entra ID 进行身份验证，无需使用个人访问令牌，简化了用户体验并提高了安全性。此集成可确保在 Power BI 中实施数据湖级别的安全控制，因而可以跨平台维护一致的安全策略。[1]

2. 其次，语义湖屋体系结构可精简数据的引入和存储。它提供了统一的存储层，支持广泛的数据格式和结构，可显著提高数据处理和转换的效率。此设置不仅简化了分析堆栈，还提高了 BI 工具的数据质量和可访问性，可直接在大型数据集上实现更复杂的数据建模和分析。[2]

3. 最后，该集成支持高级分析场景，可简化大型数据卷的管理和分析。Power BI 中的 DirectQuery 选项在这里起着至关重要的作用，允许用户在不将数据移出湖屋的情况下执行实时分析。此功能对于保持报告和仪表板的最新准确性至关重要，可为企业提供深入且立即可行的见解。[3]

这些功能共同打造了 Azure Databricks 与 Power BI 的强大组合，为企业提供了安全有效地利用其数据的高级工具。要了解有关 Power BI 与 Azure 湖屋集成的更多信息，请参阅以下文档：https://learn.microsoft.com/fabric/get-started/directlake-overview

1 利用 Microsoft Power BI 和 Azure Databricks 中的湖屋推动商业智能发展：第 1 部分 - 基础知识 - Microsoft 社区中心

2 利用 Power BI 和 Azure Databricks 中的湖屋推动发展：第 3 部分 - 调整 Azure Databricks SQL - Microsoft 社区中心

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

使用 Data Activator 通过 Power BI 提醒 Azure Databricks 中的数据变化

企业的一个常见场景是，他们希望在特定指标超过某些阈值时收到警报。例如，他们可能想知道特定商品的销售额是否突然出现意外增长，或交易量是否暴跌至正常范围以下，这表明交易管道中可能存在问题。

这些场景可以由 Fabric 中一个名为 Data Activator 的新功能来处理。这个无代码工具可监控 Power BI 报告中的数据，在数据符合特定模式或达到指定阈值时自动采取行动。当这些事件发生时，Data Activator 可以采取行动，例如提醒用户或启动 Power Automate 工作流。

若要启用 Data Activator，请按照以下官方文档操作：https://learn.microsoft.com/fabric/data-activator/

要使用 Data Activator 创建一个当 Power BI 报告中的冷冻库温度降至 30° F 以下时触发的警报，请按照以下步骤操作，以便在 Fabric 工作区内监控冷冻库温度：

1. 确认包含冷冻库温度数据的 Power BI 报告已在线发布到配备了 Premium 容量的 Fabric 工作区。

2. 选择温度视觉对象：

   a. **访问报告**：打开跟踪冷冻库温度的特定 Power BI 报告。

   b. **选择相关的视觉对象**：找到显示冷冻库温度的视觉对象。

3. 单击温度视觉对象右上角的省略号 **(...)**，选择“Set Alert”（设置警报），或使用 Power BI 工具栏中的“Set Alert”（设置警报）按钮。

4. 在“Set Alert”（设置警报）窗格中，指定接收警报的方式（电子邮件或 Teams）。如果视觉对象内包含多个冷冻库（维度），请使用“For each”（针对每个）下拉菜单，选择要监控的具体维度（冷冻库）。

3. 定义警报条件，例如温度降至 30° F 以下。Data Activator 将监控温度，并在满足此条件时发出通知。

4. 决定在 Power BI 中保存 Data Activator 触发器的位置。可将其添加到现有 Reflex 项，也可以创建一个新的项。

5. 单击“Create alert”（创建警报），完成 Data Activator 触发器。如果你希望在激活触发器之前，先在 Data Activator 中编辑该触发器，可以取消选中“Start my alert”（启动我的警报）。

执行完上述步骤后，你已成功在 Data Activator 中设置了警报，可以在受监控的冷冻库温度低于 30° F 时收到通知，根据需要立即采取行动。相关数据更新完成后，你应当能够收到之前配置的来自 Data Activator 的警报。

使用 Lakehouse Monitoring with Alerts 功能，在 Azure Databricks 中发生变化时发出警报

企业通常需要在数据质量指标超过特定阈值时收到警报。例如，他们可能想知道，特定字段内缺失值的数量是否突然出现意外激增，这表明事务管道中可能存在问题；或者，机器学习模型的预测质量是否下降，这表明需要使用更新的数据来重新训练模型。

这些场景可通过 Azure Databricks 的 Lakehouse Monitoring with Alerts 功能来处理。这个无代码工具可监控 Unity Catalog 中的数据质量，并在数据符合特定条件或超过阈值时自动采取措施。当这些事件发生时，Alerts 将执行指定的操作，例如通过电子邮件、Slack 或 Teams 发送通知。警报还可以调用 Webhook 操作，允许用户根据数据中的变化构建可扩展的自定义工作流。

监控程序是按指定计划运行的进程，用于检查特定表的数据质量。用户创建监控程序后，它会计算表的数据质量指标，并将当前值存储在一个单独的系统表中。监控程序每次运行时，都会重新计算质量指标，并将其与原始值进行比较。如果发现质量下降，就会发出警报。有关如何创建监控程序的详细信息，请参阅以下文档：https://docs.databricks.com/lakehouse-monitoring/create-monitor-ui.html

如果监控程序检测到表中数据的质量有所下降，就会引发指定警报。这可用于向数据工程团队发送通知，以便他们进一步调查。有关如何配置这些警报的详细信息，请参阅以下文档：https://docs.databricks.com/lakehouse-monitoring/monitor-alerts.html

强强联合：Azure Databricks、Unity Catalog 和 Microsoft Fabric Purview

随着分析需求的增长和数据平台演变为更复杂的系统，平台治理（数据可用性、易用性、完整性和安全性的管理）就变得至关重要。在数据湖屋体系结构中，数据治理有助于确保数据的妥善编目、分类和管理。通过实施有效的数据治理措施，组织可以妥善管理其数据，用它们来推动业务价值。

要在数据湖屋结构体系中实现有效的数据治理，就需要实施用于管理数据的策略、程序和标准。具体包括，定义数据的所有权和管理职责、建立数据质量标准，以及实施数据安全和合规性措施。要实现这些关键的数据治理能力，Azure Databricks 和 Microsoft Fabric 都需要提供强大的现代功能。

Azure Databricks 中的 Unity Catalog

Azure Databricks 中包含 Unity Catalog，后者可以为组织的数据存储位置提供集中的细粒度访问控制、数据访问审核，以及从引入到所有数据工作负载的世系跟踪，而 Azure Databricks 可提供列级和行级访问控制以及数据发现工具。它现在还包含系统表，用于提供查询审核数据、计费数据和世系的简单方法。此外，Unity Catalog 还具备 AI 支持，可自动记录表和列、促进语义搜索，帮助发掘相关数据产品。

用于治理 Microsoft Fabric 的 Microsoft Purview

Fabric 可与 Microsoft Purview 集成，以实现数据治理、信息保护和数据丢失防护等功能。借助信息保护功能，企业可以发现、分类和保护湖屋中存储的数据并对其应用敏感度标签。数据丢失防护功能使用策略来检测敏感数据何时上传到 Power BI 语义模型或其他支持的 Fabric 资产中。它还可以帮助检测常见的敏感数据。Fabric 还包含用于发现数据世系的工具，能够在数据从原始来源出发，经过各种转换，最终进入各种报告模型的过程中，通过分析过程对其进行跟踪。

用于精简数据治理的 Microsoft Purview 和 Unity Catalog

Microsoft Purview 和 Unity Catalog 是两款功能强大的工具，旨在增强云环境中的数据治理和管理，特别适合 Microsoft 提供的广泛云服务的用户。

Microsoft Purview 的广泛治理功能可扩展到 Azure Databricks 环境中，由 Unity Catalog 将特定的治理和安全措施应用于 Azure Databricks 工作区。这种集成让组织能够在所有平台上保持一致的治理策略，从而提高安全性和运营效率。组织可确保数据策略得到统一应用、数据世系清晰且可审核，整个数据资产满足所有法规遵从性要求。

Unity Catalog 提供了一个复杂的集中式治理解决方案，用于管理 Azure Databricks 湖屋平台中的各种数据资产。它与 Azure 无缝集成，可提供细粒度的治理功能，包括访问控制、审核和数据世系。Unity Catalog 可简化跨多个 Azure Databricks 工作区的数据管理，让组织能够在其数据资产（无论是文件、表还是机器学习模型）中实施一致的安全性和合规性策略。

Unity Catalog 可为数据访问策略提供单一控制点，这些策略将统一应用于所有工作区。这就确保了数据治理能够以集中的方式深度集成到 Azure Databricks 环境中，增强安全性和治理效果。此外，Unity Catalog 还支持全面的数据发现，让用户能够在遵守既定的访问控制与策略的同时，更轻松地查找和访问所需的数据。这种统一的方法有助于精简操作并降低大型和多样化数据环境管理工作中常见的复杂性。

借助 Microsoft Purview，企业可以通过 Fabric 保持对数据的控制权，从各种来源一直到详细的报告，实现数据的无缝集成和管理。除了一套用于跨不同环境保护敏感数据的工具外，Microsoft Purview 还提供了使用敏感度标签发现、分类和保护敏感数据等功能。它还支持专门为复杂环境（如 Power BI 语义模型）量身定制的全面审核和数据丢失防护策略。

最佳实践

Azure Databricks 与 Microsoft Purview 的集成旨在最大限度地改善 Azure Databricks 环境中的数据治理和安全性。适用于此集成的主要最佳实践包括：

- **安全访问关键数据**：Microsoft Purview 可用于自动发现和分类 Azure Databricks 中的数据、可视化数据世系，以及有效地管理访问控制。这可确保只有经过批准的人员才能访问敏感或关键数据，并且所有数据策略都统一应用于 Azure 服务。

- **使用两个单独的连接器来管理元数据**：Microsoft Purview 为 Azure Databricks 提供了两个单独的连接器。大多数企业都会使用 Azure Databricks Unity Catalog 连接器，因为 Unity Catalog 支持 Azure Databricks 中的许多现代功能。但是，对于尚未迁移到 Unity Catalog，仍使用 Hive 来管理元数据的客户，Microsoft Purview 提供了一个 Azure Databricks Hive 元存储连接器。

- **使用自定义规则集**：企业可以使用 Microsoft Purview 来扫描目录、架构、表和视图。作为最佳实践，企业应使用 Microsoft Purview 提供的规则集之外的自定义规则集。通过只使用特定区域所需的分类规则，可以为世界上不同的区域创建自定义规则集，从而加快扫描过程。

- **使用标记工具指明数据敏感性**：Microsoft Purview 中的标记工具可用于 Unity Catalog 数据，以指明文件和数据列的敏感度。这些标签与数据一起传播，可以由 Microsoft 数据生态系统中的其他工具（如 SharePoint 和 Power BI）使用，以自动应用数据处理策略。

Microsoft Purview、OneLake 中的 Azure 安全性和 Unity Catalog 的组合功能支持具有复原能力的敏捷数据治理策略，让企业能够在数字环境中有效地使用其数据资产。

Microsoft Fabric 中的数据工厂和 Azure Databricks 活动

Microsoft Fabric 中的 Azure Databricks 活动代表了云环境中数据处理的重大进步，可将 Azure 数据工厂的广泛功能集成到一个更加统一且稳健的框架中。借助新的 Azure Databricks 活动，用户可以轻松在 Fabric 中创建和管理数据管道，将复杂的分析和处理任务直接整合到他们的工作流中。

用户可以配置 Azure Databricks 群集，用于直接在 Fabric 内进行数据处理，就像 Azure 数据工厂提供的功能一样。这包括能够设置 Azure 现成实例，以更低的成本访问未使用的 Azure 计算容量，并指定群集策略以确保群集配置符合组织的标准和要求。

一项活动，包含所有三种作业类型：
笔记本、Jar、Python

Unity Catalog 支持和策略 ID 集成

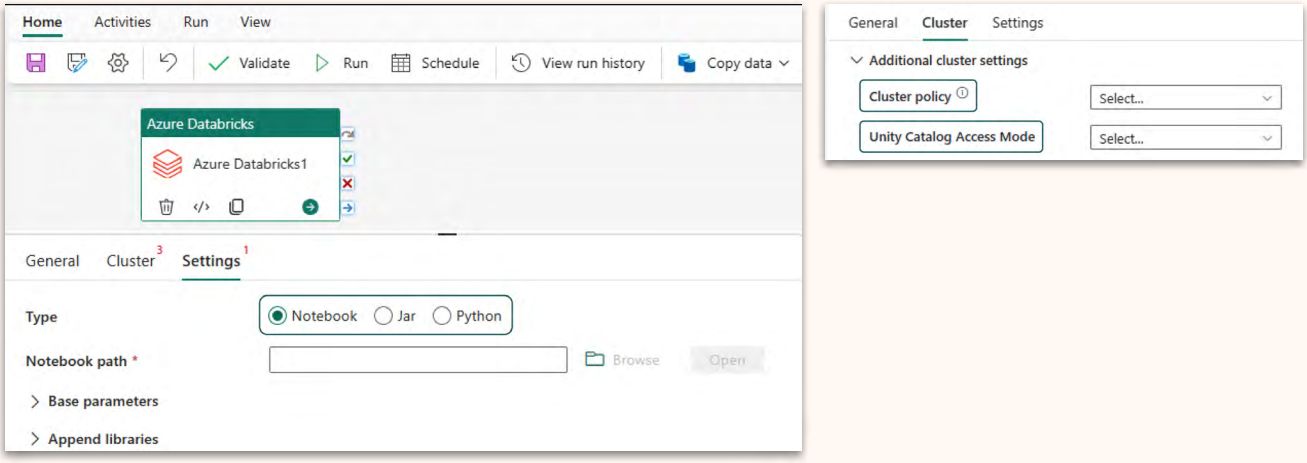


图 6：Azure Databricks 与数据工厂和 Microsoft Fabric 之间的无缝集成

此外，增强的 Azure Databricks 集成还引入了几个新功能。分别为：

- **配置 Unity Catalog 访问模式**：用户可以配置 Unity Catalog 的访问模式，通过更精细地管理权限来增强治理和安全性。
- **在单个 Databricks 活动中运行多个任务**：用户可以在单个 Databricks 活动中运行各种任务，如笔记本、JAR 和 Python 脚本，从而精简流程并降低以往管理多种数据作业时的复杂性。
- **实时监控笔记本运行**：操作期间，用户可启动数据管道并立即监控其执行，系统将提供指向 Azure Databricks 实例中输出的直接链接。这种实时监控功能允许用户直接从 Fabric 跟踪笔记本运行的详细信息，包括群集性能和计算效率。

这种集成不仅能将 Azure 数据工厂中的现有 Azure Databricks 功能引入 Fabric，还增加了新功能，如群集策略和 Unity Catalog 支持，改善了整体数据管理和分析体验。

通过生成式 AI 增强组织能力

先进的 AI 模型已经改变了技术格局。企业正努力发掘数据的潜力，并利用 AI 来扩展业务能力。包括使用生成式 AI，在 Azure 上构建数据湖屋。

接下来，我们将详细介绍如何向 ChatGPT 描述你所需的体系结构，让它生成可以在 Azure 中实现的代码。

使用 Azure OpenAI 生成体系结构和代码

可以使用提示工程来生成体系结构计划和代码，具体过程与总结技术文档一样。

提示输入

> 我想让你承担数据工程师的工作。我将详细介绍一个场景，在这个场景中，客户需要利用 `ADLS Gen2` 中存储的数据。`Azure Databricks` 和 `Microsoft Fabric` 将访问它，为机器学习模型加载数据。你的任务是构建一个能够满足这项需求的体系结构。

Azure OpenAI 的回答

> 在这个场景中，客户需要利用 `Azure Data Lake Storage Gen2 (ADLS Gen2)` 中存储的数据，而 `Azure Databricks` 和 `Microsoft Fabric` 则需要访问 `ADLS Gen2`，以便将数据加载到机器学习模型中。我们将设计一个能够有效利用这些服务的体系结构，以满足上述需求。这个体系结构旨在精简数据处理、增强数据治理，确保可扩展的机器学习操作。

体系结构概述

- 数据存储层：Azure Data Lake Storage Gen2：充当存储原始数据、已处理数据和机器学习工件的中央存储库。ADLS Gen2 针对大规模分析场景进行了优化，支持分层命名空间，简化了数据管理。
- 数据处理和分析层：Azure Databricks：用作数据处理、分析和机器学习模型开发的主要平台。Azure Databricks 提供了一个支持多种编程语言的协作环境，并与 ADLS Gen2 无缝集成，可直接访问数据。
- 数据管理和治理层：Microsoft Fabric：用于管理和治理整个数据环境中的数据。Microsoft Fabric 允许创建统一的数据治理模型，其中包括 ADLS Gen2 中的数据以及由 Azure Databricks 处理或生成的数据。
- 机器学习操作化：Azure 机器学习：与 Azure Databricks 集成，用于部署和管理机器学习模型。此服务提供用于模型版本控制、监控和服务的工具，促进了机器学习模型的操作化。

详细工作流

1. 数据引入：数据从各种来源（包括结构化数据库、IoT 设备、日志文件等）引入 ADLS Gen2。这些数据以原始格式存储在分层文件系统结构中。

> **备注**
>
> 有关 Azure OpenAI 生成的其余内容，请参见*附录*。
>
> 此处反映的提示和答案可能无法精准体现你的经历。

使用 Databricks Assistant 生成和更正代码

Databricks Assistant 是集成在 Azure Databricks 环境内且由 AI 提供支持的工具，可通过直接在 Databricks 笔记本内协助用户进行代码生成、错误解决和文档化来提高编码效率。它可利用 Azure AI 服务的功能，增强开发人员与平台内数据和代码交互的方式。

Databricks Assistant 是一个强大的工具，可通过在 Azure Databricks 工作区内自动化日常任务、优化代码、解释功能以及开展故障排查，为开发人员提供支持。这不仅能加快开发过程，还有助于保持高标准的代码质量和文档化，使其成为数据工程和分析工作流中的资产。

Databricks Assistant 可通过以下方式，利用 AI 在 Azure Databricks 环境中提供代码生成、错误解决和文档化等帮助，提高数据科学和工程方面的生产力：

- **代码生成**：Databricks Assistant 允许用户以自然语言输入要求，从而简化了编码过程。它可以生成可执行的 SQL 查询，或将代码从一种语言转换为另一种语言，例如将 Python Pandas 代码转换为 PySpark。此功能可加快开发速度并减少手动编码错误。
- **解决错误**：Databricks Assistant 可快速识别和澄清编码错误，通过生成纠正性代码片段来提供解决方案。这对新手和有经验的程序员都很有价值，因为它为常见的语法和运行时问题提供了即时解决方案，从而最大限度地减少了停机时间。
- **文档化**：它通过自动生成解释代码块功能的注释来帮助编写代码，支持维护实现项目长期可持续性和团队协作所必需的干净且可理解的代码库。
- **上下文帮助和学习**：Databricks Assistant 通过了解用户环境（包括常用的表、架构和以前的查询）来帮助提供上下文。它使用这些上下文来提供准确的答案和量身定制的代码片段，增强特定于项目的支持。

- **可视化和仪表板**：在 Lakeview 等可视化环境中，Databricks Assistant 可以根据用户提示生成数据可视化效果，无需深入的技术专业知识，即可快速创建和迭代直观显示。
- **数据引入和 ETL 流程**：Databricks Assistant 通过自动生成和优化代码，加快数据管道的设置和执行，精简数据引入和 ETL 任务。
- **安全性和合规性**：Azure Databricks 可确保与 Databricks Assistant 交互的安全性，同时遵守用户权限和数据治理策略，使其适合在敏感且受监管的环境中使用。
- **集成和可访问性**：Databricks Assistant 可通过笔记本、SQL 编辑器和文件编辑器访问，是处理各种数据任务的多功能工具。
- **反馈和迭代**：用户可直接通过平台提供反馈，帮助提高 Databricks Assistant 的准确性和功能性，从而确保该工具可以根据用户的需求和挑战不断发展。

Azure Databricks 工作区可提供为期 14 天的免费试用，包括 Assistant 的访问权限，让潜在用户无需进行初始投资，即可评估其功能并集成到自己的工作流中。

下面是一些示例，介绍如何在不同场景中使用 Databricks Assistant 提高生产力并简化任务：

1. **代码生成**

   场景：用户需要从 DataFrame 中按区域提取和汇总销售数据。

   用户输入：

   ```
   生成一个 SQL 查询，用于在 sales_data 表中，按地区对所有销售额求和。
   ```

   响应：

   ```
   sqlCopy code

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

   作用：让用户无需手动编写 SQL 查询，即可快速获取所需代码。

2. **修复错误**

场景：用户编写的 PySpark 代码由于语法错误而失败。

用户输入：

> 这些代码不起作用。
> 能帮忙修复吗?

诊断和修复：Databricks Assistant 在 DataFrame 操作中发现缺少了一个逗点，给出了代码修改建议并突出显示了所做更改。

作用：用户可获得即时反馈和更正建议，帮助加快故障排查速度并减少挫折感。

3. **代码文档化**

场景：开发人员希望为复杂函数添加注释，改善代码的可读性。

用户输入：

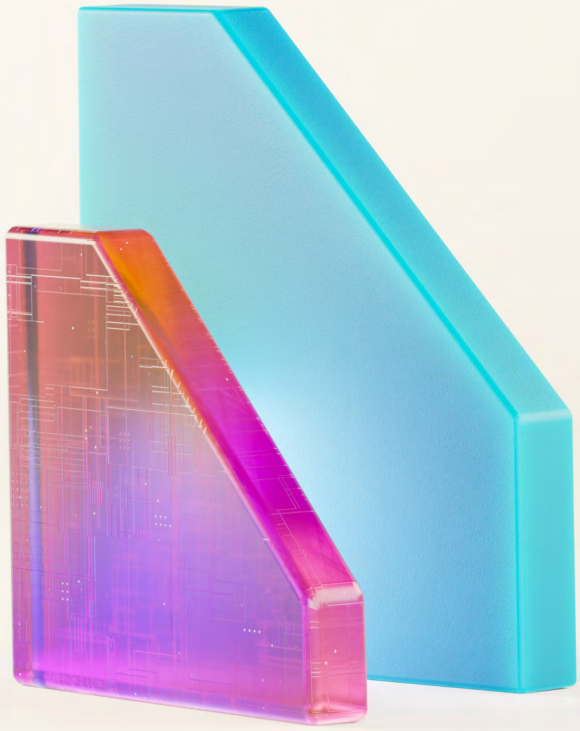
> 能否为这个函数添加注释，详细解释每个步骤?

响应：Databricks Assistant 可在每个重要的代码行或代码块之前添加注释，解释其功能，如初始化变量、错误处理和逻辑流等。

作用：确保代码易于理解，可供将来参考或供其他团队成员使用，从而增强可维护性。

以上示例说明了 Databricks Assistant 在实际开发环境中带来的切实好处：不仅能精简编码过程、简化错误解决过程，还能确保进行全面的文档记录。

> **备注**
> 此处反映的提示和答案可能无法精准体现你的经历。



通过各种实践案例探索实际应用

在前面的示例中，你使用 Python 代码来读取数据并将其汇总在一起以回答一些业务问题。本节将介绍一种能够替代 Python 代码的代码读取方法，以及如何利用 AI，让企业用户能够使用英语（而不是查询语言）来查询湖屋数据。

使用 English SDK for Spark 在 Azure Databricks 和 Fabric 中编写查询

要使用 English SDK for Apache Spark，首先应满足以下要求：

> **备注**
> Azure Databricks 建议使用 GPT-4。

1. **安装 English SDK 包**：首先将 SDK 添加到你的环境中。在笔记本中使用 `%pip install pyspark-ai --upgrade` 命令，以确保拥有最新版本。
2. **重新启动 Python 内核**：安装完成后，需要重新启动 Python 内核以应用更新。在新单元格中执行 `dbutils.library.restartPython()` 以重置环境。
3. **设置 OpenAI API 密钥**：进行身份验证必须要使用 OpenAI 的 API 密钥。具体做法是，使用 Python 代码 `os.environ['OPENAI_API_KEY'] = '<your-openai-api-key>'` 设置环境变量，用你的实际 API 密钥替换 `<your-openai-api-key>`。
4. **激活 SDK**：要使用 SDK，请在笔记本中激活它。这包括使用首选语言模型（如 GPT-4）初始化 SDK，然后将其激活以开始解释英语查询。
5. **创建 DataFrame**：在笔记本中使用 SQL 查询，以便从 Azure Databricks 工作区中获取数据，然后将其保存为 DataFrame。这个 DataFrame 将成为你进行英语查询的基础。

6. **使用英语进行查询**：最后，用简单的英语问题来查询 DataFrame。SDK 将解释这些问题并执行相应的 SQL 查询，然后将结果直接返回给笔记本。

下面是一个在 English SDK for Apache Spark 中使用英语查询的示例：

```
2016 年 1 月，每天平均的出行距离是多少？请将平均值近似到十分位。
```

这个查询演示了如何借助 English SDK，用简单的英语进行数据分析活动，例如计算数据集中的平均值，从而允许 Apache Spark 解释和执行英语指令。

下面是另一个在 English SDK for Apache Spark 中使用英语查询的示例：

```
显示上一季度每个产品类别的总收入。
```

这种类型的查询说明了用户如何使用自然语言进行查询，以了解指定时间段（如上一季度）内按类别细分的特定财务指标（如总收入）。这种方法可以将复杂的数据分析任务简化为简单直接的英语问题。

在 Microsoft Fabric 中创建笔记本

Fabric 笔记本是构建 Apache Spark 作业和开展机器学习实验的关键工具。凭借对高级可视化效果和 Markdown 文本集成的支持，它提供了一个基于 Web 的交互式平台，很多数据科学家和工程师编码时都会用到它。数据科学家依靠这些笔记本来开发和部署机器学习模型，包括实验、模型跟踪和部署阶段。Fabric 笔记本的特性如下：

- 无需设置即可使用
- 可提供用于数据探索和处理的直观、低代码界面
- 可通过集成的企业级功能增强数据安全性
- 可利用 Spark 的强大功能分析各种格式的数据（包括 CSV、TXT、JSON、Parquet 和 Delta Lake）

创建笔记本

创建笔记本时，用户有两个选项：创建新笔记本或导入现有笔记本。组织可以按照熟悉的 Fabric 项创建工作流，创建一个新的笔记本：

1. 直接通过 Fabric 的“Data Engineering”（数据工程）或“Data Science”（数据科学）主页，或工作区的“New”（新建）选项，启动一个新笔记本。
2. 在同一窗口中选择“**Import Notebook**”（导入笔记本），可以导入现有笔记本，如 Azure Databricks 笔记本文件。
3. 打开笔记本后，可以向其中添加代码，以将数据写入 OneLake。

在 OneLake 中处理数据非常简单，不需要复杂的设置即可访问数据。

通过 Microsoft Fabric 数据工程笔记本将数据加载到 OneLake 中

```
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)
```

读取和分析数据

数据上传成功后，尝试读取和分析数据：

```
# 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 中通过 Parquet 创建和修改 Delta 表并在 Fabric 中反映所做更改

Azure Databricks 与 Fabric 可提供一个数据湖屋环境，允许企业使用不同的工具同时访问和分析数据。这支持对同一组数据进行广泛的数据处理活动，让组织能够更加轻松有效地管理信息并从中获得见解。

1. 在你选择的浏览器中打开 Azure Databricks 工作区，启动一个新的 Azure Databricks 笔记本。

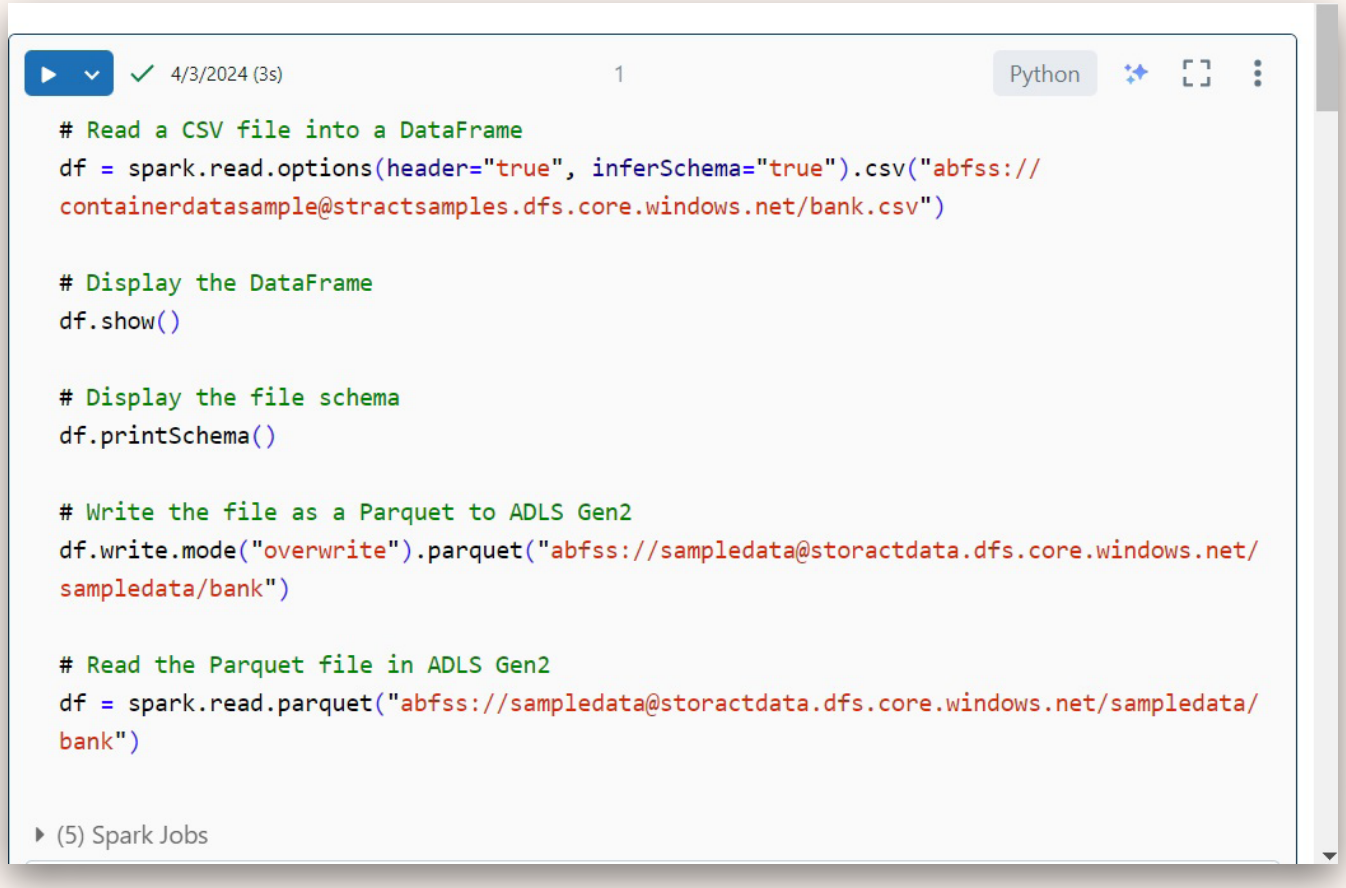


图 7：笔记本示例

2. 将以下脚本复制并粘贴到你的新笔记本中。然后，在笔记本中执行以下 Python 脚本，以便在你的 ADLS Gen2 帐户中创建一个 Delta 表。这个脚本将读取一些 Parquet 示例数据，然后将其作为 Delta 表写入你的 ADLS 帐户：

```
#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/")
```

当然，Azure Databricks 也可以读取 ADLS 帐户中的数据。

3. Azure Databricks 还可以修改先前使用 Fabric 创建的数据集。具体操作为，向你在 OneLake 中创建的 Delta Lake 表内添加一些新行：

```
# 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()
```

如上例所示，如果数据湖屋构建时所在的基础平台具有开放平台的优势，就能让企业使用各种引擎，同时处理同一组数据。

Power BI 内的 Azure Databricks 连接器

用于 Azure Databricks 的 Power BI 连接器可实现 Power BI 与 Azure Databricks 之间的无缝集成，让组织能够轻松连接、分析和可视化存储在 Azure Databricks 中的数据。这种集成支持 Microsoft Entra ID 身份验证，无需管理员生成用于连接的个人访问令牌。它旨在改善数据连接和分析体验，允许直接通过数据湖实现安全高效的数据可视化效果。

1. 获取用于在 Power BI 中设置连接的 Azure Databricks 服务器主机名和 HTTP 路径。
2. 启动 Power BI Desktop。
3. 从主屏幕中选择“Get Data”（获取数据），或导航到“File”（文件）> “Get Data”（获取数据）。
4. 搜索 **Azure Databricks**。
5. 选择“Azure Databricks connector”（Azure Databricks 连接器），然后单击“Connect”（连接）。
6. 输入你之前获得的服务器主机名和 HTTP 路径。
7. 在“Import”（导入）和“DirectQuery”模式之间选择你的数据连接方式。如果想进一步了解这些选项，请参阅 **Power BI 中的 DirectQuery** 使用文档。
8. 选择首选的身份验证方法：
    a. **个人访问令牌**：输入你的 Azure Databricks 个人访问令牌。
    b. **Microsoft Entra ID**：选择 **“Sign in”（登录）**并按提示操作。
    c. **用户名 / 密码**：此选项通常不适用。
9. 身份验证结束后，Power BI 将显示“Navigator”（导航器）窗口。你可以在这里选择要查询的 Azure Databricks 数据。如果你的工作区启用了 Unity Catalog，则首先选择目录，然后选择架构和表。

如果工作负载需要 Azure Databricks 提供的处理能力和灵活性，企业可以将 Power BI 的高级可视化功能和 Azure Databricks 一起使用。

通过 Azure Databricks 与 Microsoft Fabric 实现卓越成效

Azure Databricks 与 Microsoft Fabric 的集成提供了在现代云环境中管理和分析数据的变革性方法。Azure Databricks 为数据处理和 AI 驱动的分析提供了一个高性能平台，Fabric 则通过强大的数据管理工具增强了这些功能。这种组合让组织能够更有效地利用高级分析和 AI 解决方案。

Azure Databricks 可与 Fabric 协同工作，共同提供无缝的数据访问服务，无需冗余数据副本，从而精简了分析工作负载。这种集成支持来自 Power BI 的直接查询，因此可提高性能并简化数据体系结构。

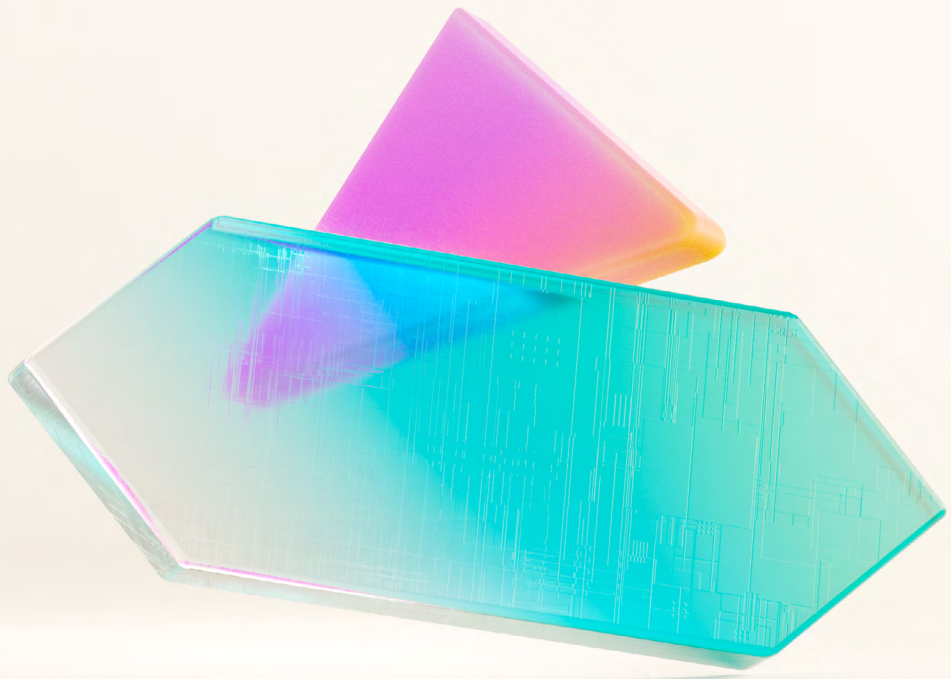
Azure Databricks 与 Fabric 之间的协同作用可释放湖屋体系结构的全部潜力，增强更有效管理、处理和分析大型数据集的能力。让组织能够轻松部署 AI 解决方案并应对复杂的分析挑战。

此外，Azure Databricks 与 Fabric 的组合功能可满足 AI 和机器学习项目在整个生命周期（从数据引入和存储到分析和部署）内的需求。这种集成可帮助组织保持对数据流程的控制，同时确保 AI 计划既可以扩展，又符合行业法规。

借助 Azure Databricks 与 Fabric 的强大集成，组织可以改善数据治理、提高运营效率并推动 AI 和分析方面的创新，让自己在面对数据驱动型经济时做好充足的准备，有能力实现战略目标。

后续行动

- **与销售专家交谈。**
- **利用 Microsoft Fabric 从数据中获得更多价值。**
- **借助 Azure Databricks 从数据中获取见解并构建 AI 解决方案。**
- **观看网络研讨会，详细了解如何让 Azure Databricks 与 Microsoft Fabric 更好地发挥协同作用。**



附录

提示输入

我想让你承担数据工程师的工作。我将详细介绍一个场景，在这个场景中，客户需要利用 ADLS Gen2 中存储的数据。Azure Databricks 和 Microsoft Fabric 将访问它，为机器学习模型加载数据。你的任务是构建一个能够满足这项需求的体系结构。

Azure OpenAI 的回答

在这个场景中，客户需要利用 Azure Data Lake Storage Gen2 (ADLS Gen2) 中存储的数据，而 Azure Databricks 和 Microsoft Fabric 则需要访问 ADLS Gen2，以便将数据加载到机器学习模型中。我们将设计一个能够有效利用这些服务的体系结构，以满足上述需求。这个体系结构旨在精简数据处理、增强数据治理，确保可扩展的机器学习操作。

体系结构概述

- 数据存储层：Azure Data Lake Storage Gen2：充当存储原始数据、已处理数据和机器学习工件的中央存储库。ADLS Gen2 针对大规模分析场景进行了优化，支持分层命名空间，简化了数据管理。
- 数据处理和分析层：Azure Databricks：用作数据处理、分析和机器学习模型开发的主要平台。Azure Databricks 提供了一个支持多种编程语言的协作环境，并与 ADLS Gen2 无缝集成，可直接访问数据。
- 数据管理和治理层：Microsoft Fabric：用于管理和治理整个数据环境中的数据。Microsoft Fabric 允许创建统一的数据治理模型，其中包括 ADLS Gen2 中的数据以及由 Azure Databricks 处理或生成的数据。

- 机器学习操作化：Azure 机器学习：与 Azure Databricks 集成，用于部署和管理机器学习模型。此服务提供用于模型版本控制、监控和服务的工具，促进了机器学习模型的操作化。

详细工作流

1. 数据引入：数据从各种来源（包括结构化数据库、IoT 设备、日志文件等）引入 ADLS Gen2。这些数据以原始格式存储在分层文件系统结构中。

2. 数据处理和探索：

   - Azure Databricks 笔记本用于访问存储在 ADLS Gen2 中的原始数据。数据工程师和科学家使用 Spark 来处理和探索这些数据，执行数据清理、转换和特征工程等任务。

   - 处理后的数据能够以经过优化的格式（如 Delta Lake）存储回 ADLS Gen2，以供进一步分析或机器学习。

3. 数据治理和世系：

   - Microsoft Fabric 被配置为对 ADLS Gen2 中的数据资产和 Azure Databricks 生成的数据产品进行编目。它提供了一个统一的数据视图，强制执行策略、数据质量规则和安全措施。

   - Microsoft Fabric 支持设置数据世系和跟踪，以便在整个数据生命周期中持续了解数据的转换和使用情况。

4. 机器学习模型的训练与部署：

   - 数据科学家使用 Azure Databricks 开发和训练机器学习模型，利用集成的 MLflow 进行实验跟踪和模型管理。

   - 经过训练的模型在 Azure 机器学习中注册，随后作为 Web 服务或容器部署到生产中，进行实时或批量评分。

5. 监控与管理：

- Azure 机器学习提供了模型监控功能，可跟踪模型的性能和数据随时间变化的偏移情况。这有助于在模型过时时重新训练工作流。
- Microsoft Fabric 可监督治理和合规性方面，确保数据访问和模型部署符合组织与监管标准。

这个体系结构为 Azure Databricks 与 Microsoft Fabric 中的 ADLS Gen2 数据利用提供了全面的解决方案，促进了高效的机器学习模型开发、治理和运营。它可确保可扩展性、安全性和合规性，满足现代数据驱动型组织的动态需求。

> **备注**
> 此处反映的提示和答案可能无法精准体现你的经历。

提示输入

基于以上信息，能否提供包含代码的详细构建说明?

Azure OpenAI 的回答

创建一个包含代码的详细构建指南，介绍如何将 Azure Data Lake Storage Gen2 (ADLS Gen2)、Azure Databricks 和 Microsoft Fabric 集成到一个用于机器学习的统一体系结构中，需要几个步骤。下面，我们将概述这个过程中的关键阶段，包括设置、数据处理和机器学习模型生命周期管理。本指南假设你对 Azure 服务具有基本的了解，并拥有可部署这些服务的订阅。

1. 设置和配置

Azure Data Lake Storage Gen2

1. 创建 ADLS Gen2 存储帐户：
   - 导航到 Azure 门户。
   - 创建一个新的存储帐户，选择你的订阅、资源组和“StorageV2”帐户类型。
   - 启用分层命名空间以使用 ADLS Gen2 功能。
2. 配置存储帐户访问权限：
   - 在存储帐户中，转到“Access keys”（访问密钥）部分，记下连接字符串以供以后使用。
   - 也可以设置 Microsoft Entra ID 身份验证以进行更安全的访问。

Azure Databricks

1. 创建 Azure Databricks 工作区：
   - 在 Azure 门户中，创建一个新的 Azure Databricks 服务。
   - 部署完成后，导航到 Azure Databricks 工作区并启动工作区。
2. 在 Azure Databricks 中创建一个群集：
   - 在 Azure Databricks 工作区中，转到“Compute”（计算）。
   - 创建一个新群集，确保它与计划使用的库兼容。
3. 将 ADLS Gen2 装入 Azure Databricks（用于直接访问数据）：
   - 在 Azure Databricks 笔记本中，使用以下代码装载 ADLS Gen2 存储以进行直接访问：

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)
```

- 将占位符替换为实际的 Microsoft Entra ID 应用程序注册详细信息和 ADLS Gen2 存储详细信息。

Microsoft Fabric

1. 设置 Microsoft Fabric：

   - Microsoft Fabric 目前还是一个概念性集成点，可能未公开提供直接面向客户的设置说明。设置过程通常会涉及环境配置，以便在 Azure Databricks 和 ADLS Gen2 中对数据资产进行编目和管理。

   - 出于演示目的，我们将继续以下步骤，假设 Microsoft Fabric 用于治理和保护数据访问，确保合规性和数据质量。

2. 使用 Azure Databricks 进行数据处理

   1. 从 ADLS Gen2 加载数据：

      - 使用安装路径将数据读取到 Azure Databricks DataFrame 中：

        pythonCopy 代码

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

   2. 数据转换：

      - 使用 Spark DataFrame 操作执行必要的数据转换。

3. 保存处理后的数据：

   - 将处理后的数据以经过优化的格式（如 Delta Lake）保存回 ADLS Gen2：

     pythonCopy 代码

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

3. 机器学习模型的训练与部署

   4. 模型训练：

      - 使用 Azure Databricks，用处理后的数据训练机器学习模型。
      - 使用 Spark MLlib 进行逻辑回归的示例：

        pythonCopy 代码

        ```
        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")
        ```

   5. 模型部署：

      - 与 Azure 机器学习集成，以进行模型部署、监控和管理。这部分很复杂，通常涉及注册模型、创建评分脚本以及将模型部署为 Web 服务。

> **备注**
> 此处反映的提示和答案可能无法精准体现你的经历。