**Case Study: End-to-End Lakehouse Governance using Microsoft Purview**

**Objective:**  
Implement a complete data governance lifecycle on a Lakehouse architecture using Microsoft Purview. The use case involves scanning a Delta Lake, auto-classifying PII data, linking business glossary terms, and visualizing lineage from source to Power BI.

**Background:**

An e-commerce enterprise ingests transactional and user data into an Azure Data Lakehouse (using Azure Data Lake Storage Gen2 + Delta Lake format). The data includes customer profiles, order details, and browsing behavior. The enterprise wants to:

* Catalog all datasets
* Automatically classify sensitive data (PII)
* Apply consistent business definitions via glossary terms
* Track data movement from raw zone to reporting layer in Power BI

**Architecture Overview:**

* **Data Source:** ADLS Gen2 with Delta Tables (Bronze, Silver, Gold layers)
* **ETL Pipeline:** Azure Data Factory + Azure Databricks
* **Governance Layer:** Microsoft Purview
* **Reporting Layer:** Power BI

**Step-by-Step Implementation**

**1. Set Up Microsoft Purview**

* Create a Purview account.
* Define collections based on departments: e.g., Sales, Marketing, Product.
* Assign RBAC roles:
  + Collection Admin: Data Governance Lead
  + Data Curator: Domain-specific Data Stewards
  + Data Reader: Business Users

**2. Register and Scan the Lakehouse Source**

* Register ADLS Gen2 as a data source in Purview.
* Set up a scan rule set:
  + Schedule: Daily at midnight
  + File types: parquet, csv, delta
  + Exclude raw logs folder
* Use managed identity for authentication (via linked service)

**3. Enable Auto-Classification**

* Enable classification rule set during the scan.
* Use built-in PII classifiers:
  + Email Address
  + SSN
  + Credit Card
  + Date of Birth
* Post-scan, validate classified assets in the **Classifications** tab.

**4. Create & Link Glossary Terms**

* Define glossary terms like:
  + "Customer ID" (Unique identifier for customer)
  + "Order Amount" (Total value of transaction)
  + "Retention Score" (Marketing engagement metric)
* Link glossary terms to relevant columns in scanned tables via Purview UI.
* Enable term propagation across schema levels.

**5. Ingest Data with Lineage Tracking**

* In ADF:
  + Use Copy activity from Raw (Bronze) to Refined (Silver) zones.
  + Enable lineage capture by integrating ADF with Purview.
* In Databricks:
  + Use Spark jobs to create Gold layer from Silver.
  + Install and configure purview-spark-lineage jar.
  + Export lineage to Purview using REST API or Unity Catalog integration.

**6. Integrate Power BI for Lineage**

* Register Power BI workspace in Purview.
* Enable Power BI tenant-level integration.
* Scan Power BI datasets and reports.
* Validate lineage from source file to Power BI dashboard in Purview's **Lineage View**.

**Outcomes & Benefits:**

* **Visibility:** Complete traceability from raw data to final dashboard.
* **Risk Management:** Identified PII fields and enforced tagging policies.
* **Consistency:** Unified business terms across departments.
* **Compliance:** Audit-ready data inventory with lineage and classifications.

**Stretch Tasks (Advanced):**

* Automate scan configuration using DevOps pipeline (Purview REST APIs).
* Add custom classification rules (e.g., for internal IDs).
* Export lineage graph to JSON for integration into external catalogs.

**Key Tools Used:**

* Microsoft Purview
* Azure Data Factory
* Azure Databricks
* Delta Lake
* Power BI
* Azure DevOps