End-to-End On-Premises Migration

Agenda

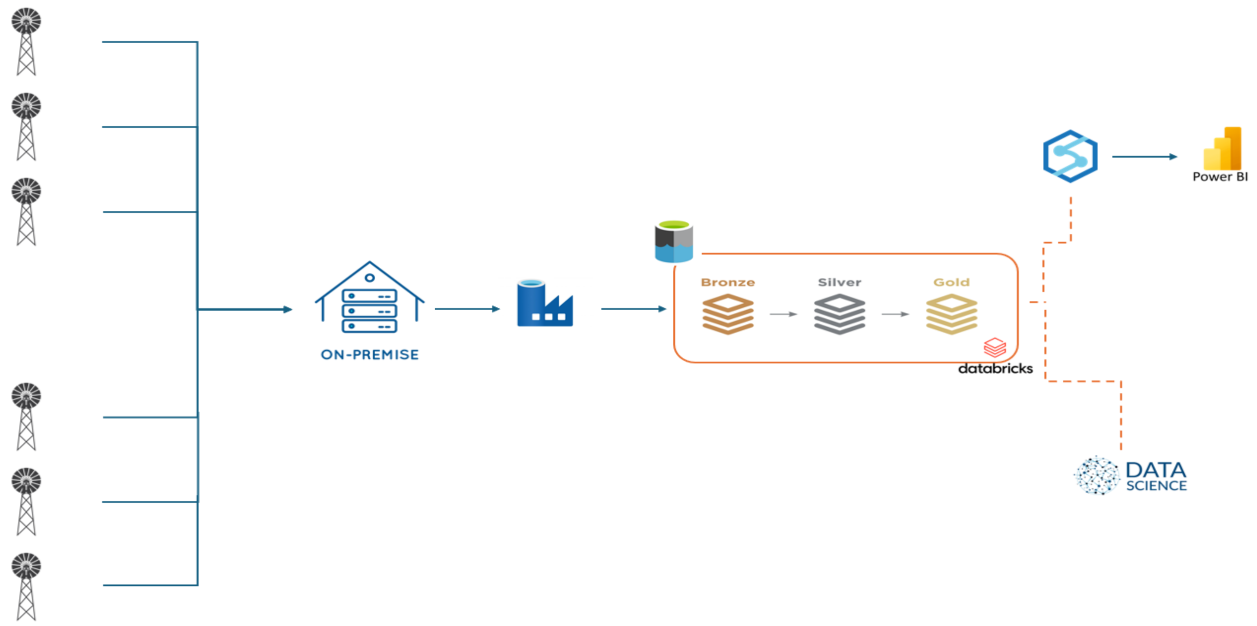
- Project Architecture
- Part 1: Environment Setup
- Part 2: Data Ingestion
- Part 3: Data Transformation
- Part 4: Data Loading
- Part 5: Data Reporting
- Part 6: End To End Pipeline

1. Project Overview

This project provides end-to-end data engineering with a range of Azure services. The main purpose is to move data from on-premises database to Azure storage, perform comprehensive data transformations with Azure Databricks, and create insightful visual reports with Power BI. This project is a learning aid for data engineering practical principles and implementation with Azure ecosystem.

2. Architecture Diagram

A visual architecture diagram outlining how the Azure tools (Azure Data Factory, Azure Databricks, Azure Data Lake Storage, Azure Synapse Analytics, and Power BI) work with each other and where they fit into data flow is highly recommended. *Such diagrams can be drawn with tools like Lucidchart or draw.io.*



3. Tools and Technologies

- **Azure Data Factory (ADF):** Cloud data movement, data integration, and ETL process service.

- **Azure Synapse Analytics:** It is a powerful analytics service that combines enterprise SQL data warehousing, big data processing, and data integration.
- **Azure Databricks:** Shared, Apache Spark-based analytics service for data processing and data transformation at scale.
- **Azure Data Lake Storage (ADLS):** Cloud storage for structured, semi-structured, and unstructured data, with high scalability and security.
- **Azure Active Directory (AAD):** Provides end-to-end identity and access management (IAM) capabilities.
- **Azure Key Vault:** Securely stores sensitive information such as credentials, API keys, and certificates.
- **Power BI:** Business data analysis tool for data visualization and interactive reporting.
- **Dataset**: AdventureWorksLT2017 (Sample dataset for practical demonstrations).

4. Environment Setup

To securely connect and transfer data from the on-premises environment to Azure:

- **Database Access Setup**: Create database user accounts and role assignments for secure and appropriate access to tables that are needed.
- **Credential Security**: Store database credentials (username and password) securely in Azure Key Vault to ensure sensitive data is protected.
- **Integration Runtime**: Create a Self-Hosted Integration Runtime within Azure Data Factory for secure connectivity from your on-premises database to Azure.
- **Data Migration**: Use Azure Data Factory pipelines to selectively ingest tables from on-premises database into Azure Data Lake Storage.

5. Data Ingestion

Data ingestion involves securely mounting Azure Data Lake Storage to Databricks:

- **App Registration**: Register your application with Azure Active Directory for managing data access.
- **Credential Passthrough Setup**: Configure Credential Passthrough for Azure Databricks clusters, eliminating the need for manual handling of sensitive credentials.
- **Data Lake Mounting**: Mount ADLS containers securely, directly, and optimally from within Databricks notebooks for analysis and transformation.

6. Data Transformation

The transformation process improves data quality and prepares it for analysis:

- **Standardized Date Formatting**: Convert all DateTime fields to a consistent, standardized date format (YYYY-MM-DD).

- **Column Pruning**: Identify and remove unnecessary columns that are not valuable for analysis to enhance performance and storage efficiency.
- **Null Value Management**: Handle null values systematically by replacing numeric nulls with zeros or mean values, and textual nulls with meaningful defaults like "Unknown."
- **Deduplication**: Implement checks to identify and remove duplicate records, ensuring data accuracy and reliability.
- **Standardization and Normalization**: Standardize units, date formats, and text casing to ensure consistency across datasets.
- **Data Integration**: Merge data from various sources, carefully resolving any conflicts to produce a coherent and unified dataset.

7. Data Loading

After transformation, data is stored efficiently for further analytics:

- **Parquet Storage**: Save transformed data in Parquet format on ADLS for optimized query performance and storage efficiency.
- **Data Warehousing**: Load data into Azure Synapse Analytics, which is optimized for reporting and analytical queries, allowing seamless integration with analytics tools.

8. Data Reporting

Power BI serves as the final stage of data visualization and reporting:

- **Data Connection**: Establish connections from Power BI to Azure Synapse or directly to Azure Data Lake Storage.
- **Data Modeling**: Develop an efficient star schema or semantic model to facilitate easy exploration and reporting within Power BI.
- **Dashboard Creation**: Develop interactive and visually appealing dashboards tailored to business stakeholders' needs, enabling easy interpretation of data insights.

9. End-to-End Pipeline Summary

- Extract data from an on-premises database using Azure Data Factory.
- Securely ingest data into Azure Data Lake Storage.
- Transform and cleanse data using Apache Spark in Azure Databricks.
- Store transformed data in optimized Parquet format.
- Load data into Azure Synapse Analytics for effective querying.
- Visualize insights using Power BI dashboards, ensuring business stakeholders can interpret and act upon data insights effectively.

10. Best Practices

- **Comprehensive Documentation**: Maintain detailed documentation of each step, from initial configurations to data transformations.
- **Pipeline Monitoring and Quality Assurance**: Regularly monitor pipeline executions and implement automated quality checks to ensure data accuracy.
- **Security Compliance**: Always secure sensitive information using Azure Key Vault and adhere to Azure AD best practices for access management.
- **Optimization**: Optimize data storage and data models for performance, scalability, and cost-efficiency.
- **Automation**: Automate repetitive tasks to minimize human errors and improve operational efficiency.