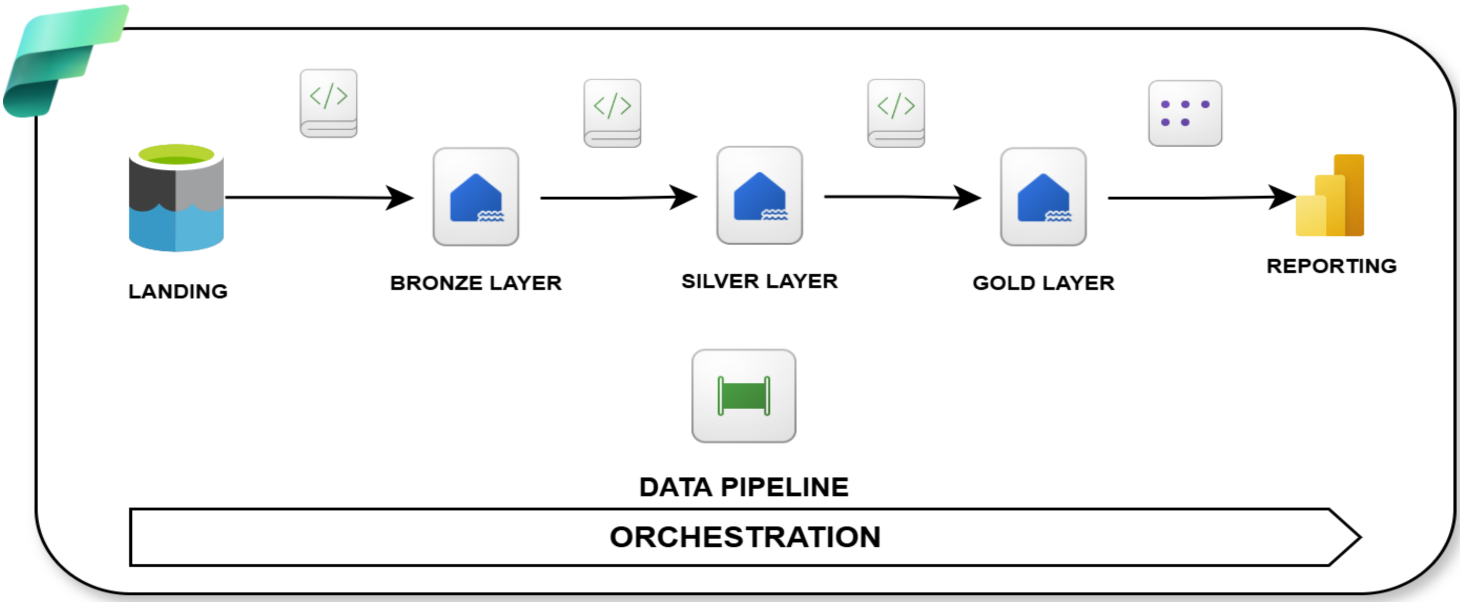
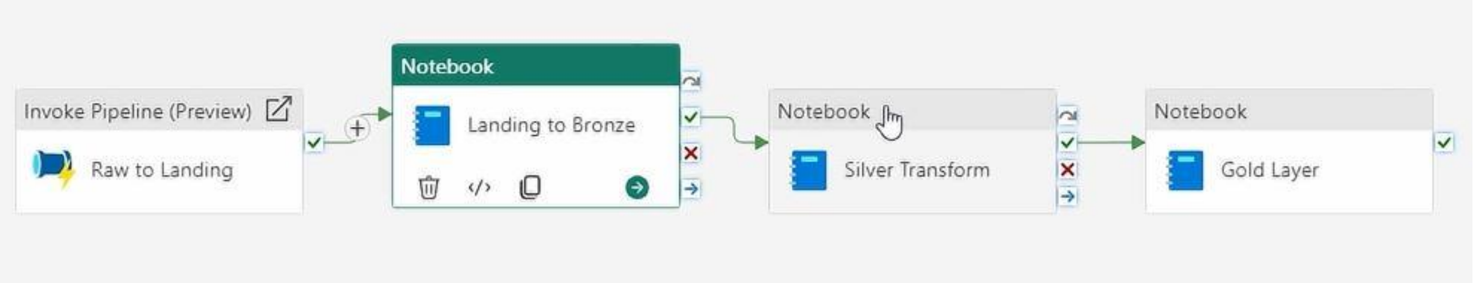
End to End Data Solution in Microsoft Fabric using the Medallion Architecture

Overview

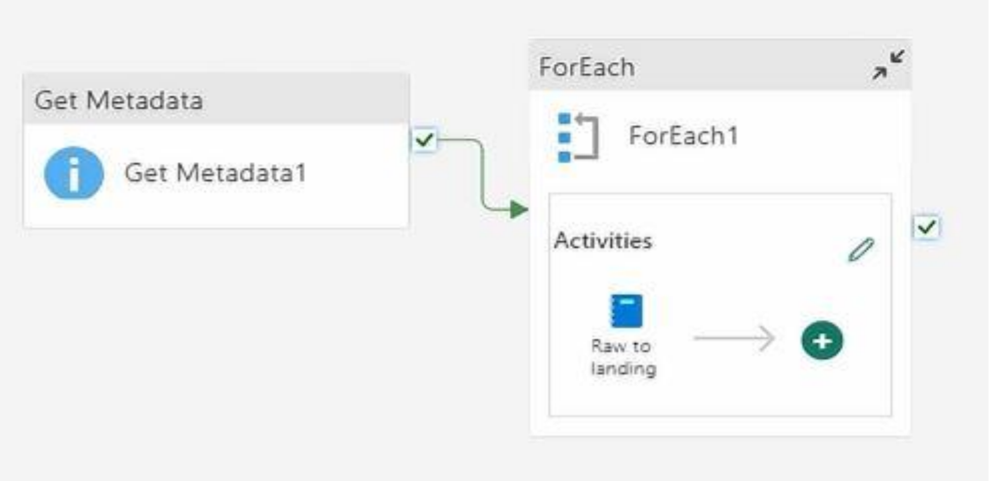
- The medallion architecture is a commonly used data Lakehouse implementation that provides a structured approach to managing data quality and optimizing performance at every stage of the data pipeline.
- The goal is to implement this architecture in Microsoft Fabric, starting from data ingestion to transformation and finally serving the insights to the reporting layer.
- The project will simulate data generated from an online learning platform, including student information, course details, enrollment data, and assignment scores.
- The data will be stored in Azure Data Lake Gen2 initially, then ingested incrementally into a Fabric Lakehouse, progressing through bronze, silver, and gold layers.



Complete Pipeline Orchestration:



Raw to Landing Pipeline:



Medallion Architecture Implementation in Fabric

There are multiple ways to implement the medallion architecture in Fabric:

1. **Single Lakehouse with Folders**: Create one lakehouse and divide it into three folders (Bronze, Silver, Gold). This is suitable for small teams and use cases.
2. **Multiple Lakehouses within One Workspace**: Create separate lakehouses for Bronze, Silver, and Gold within a single workspace. This provides a clear distinction between layers and is useful for handling multiple data types. This approach is used in the end-to-end project.
3. **Separate Workspaces**: Create multiple workspaces to hold each layer (Silver, Bronze, Gold). This is useful if there's a specific client requirement to handle this.

Tools Used in Medallion Layers

- **Bronze Layer**: Ingests data as-is (raw or landing layer data in Delta or Parquet format). Tools: Data pipelines, data flows, or notebooks.
- **Silver Layer**: Data cleaning, data validation, and business transformations. Tools: Data flows (visual transformations) or notebooks.
- **Gold Layer**: Additional transformations (if needed) and data modeling (facts and dimension tables). Tools: SQL endpoint or semantic model. End users can work in Power BI.

Project Architecture

1. **Data Generation**: Simulate LMS datasets of students, including information on courses, enrollment, and assignments.
2. **Landing Zone**: Store the dataset in Azure Data Lake Gen2 storage. Data will be ingested incrementally.
3. **Bronze Layer**: Ingest data from the landing zone to the bronze layer in the Lakehouse. The bronze layer will have the same raw data as the landing zone but in a better compressed format.
4. **Silver Layer**: Clean the data from the bronze layer and perform business transformations.
5. **Gold Layer**: Create fact and dimension tables in a Lakehouse.
6. **Reporting**: Use the data for reporting purposes by creating a semantic model in Power BI. Data scientists can also use the data for analysis.
7. **Automation**: Use Microsoft Fabric notebooks and data factory pipelines to implement and orchestrate the solution, making it data-driven.

Setting Up the Project

1. **Create a Workspace**: Create a separate workspace in Fabric for the project (e.g., "Fabric dev").
2. **Data**: The data set contains the information of students who have enrolled to the courses along with the course ID, enrollment date, the time they have spent on the courses, assignments taken and the assignments course. The data includes student ID, name, gender, grade, course ID, course name, enrollment/completion dates, final grade, attendance, time spent, quiz/assignment details, project score, access method (mobile/internet), parent involvement, and feedback score. There are 31 columns in total.
3. **Raw and Landing Folders**: Create 'raw' and 'landing' folders in Azure Data Lake Gen2. The raw folder will simulate data coming from a website. The data will be uploaded manually into the raw folder as a CSV file. The landing folder will store the data after it has been processed.

Moving Data from Raw to Landing

1. **Data Upload**: Upload data to the raw layer in the format `LMS_YYYYMMDD.csv`. Only one file is expected per day.
2. **Incremental Ingestion**: Create a pipeline to incrementally take data from the raw folder to the landing zone.
3. **Partitioning**: Partition the data in the landing zone based on the processing date. Create a process date column.

Steps to Move Data

- Data comes to the raw layer in `LMS_YYYYMMDD` format.

- Upload the file to the raw folder.
- Incrementally take the data and write it to the landing zone.
- Partition the data based on the processing date (the date when the data is processed into the landing zone).

Incremental Ingestion Types

- **Timestamp-based**: Ingest data based on a timestamp column (last updated date, created date, inserted date).
- **Change Data Capture (CDC)**: Capture changes (inserts, updates, deletes) using database logs or triggers.
- **Delta Ingestion**: Use a reference column (primary key) to track changes.
- **Batch Window Ingestion**: Ingest data based on a fixed time period window (hourly, daily, weekly).
- **Partition-based Ingestion**: Ingest data based on partitions (date, region).

The selection of an incremental load depends on the data type, data sources, frequency of updates, and data needs for reporting. This project uses partition-based and batch window-based incremental loads.

Performing Incremental Loading

1. **Parameterization**: Parameterize the notebook to accept the file name and processing date as inputs.
2. **Data Pipeline**: Create a data pipeline to automate the data ingestion process.
3. **Get Metadata**: Use the "Get Metadata" activity to get the metadata of the file in the raw folder.
4. **For Each Activity**: Use a "For Each" activity to iterate through each file in the raw folder.
5. **Notebook Activity**: Call the notebook within the "For Each" activity. Pass the file name and processing date as parameters to the notebook.
6. **Connection**: Create a connection to Azure Data Lake Gen2.
7. **Testing**: Test the pipeline by uploading a new file to the raw folder and running the pipeline.
8. **Scheduling**: Schedule the pipeline to run daily.

Taking Data from Landing to Bronze

1. **Create Lakehouse**: Create a lakehouse for the bronze layer (e.g., `LH_bronze`).
2. **Notebook**: Use a Fabric notebook to ingest data from Azure Data Lake to the bronze layer.
3. **Processing Date**: Take only today’s date data by comparing it against the processing date data.
4. **Bronze Table**: Write the data to the bronze table.
5. **Upsert Logic**: Handle repeating or updated data by implementing upsert logic. Only changed records will be updated. New records will be inserted.

Implementing Upsert Logic

- Create a notebook.
- Pick data from the landing folder by checking against today’s date.
- Incrementally ingest the data to the bronze table.

Data Cleaning and Transformation for Silver Layer

- **Data Cleaning**: Ensure data is cleaned and of high quality.
  - Handle duplicates.
  - Handle missing or null values.
  - Standardize date formats.
  - Maintain logical consistency (e.g., completion date > enrollment date).
- **Business Transformations**: Perform business transformations.
  - Completion time days = completion date - enrollment date.
  - Performance score = weightage based on quizzes, assignments, and project score.
  - Course completion rate (on-time/delayed).
- **Upsert Logic for Silver**: Implement upsert logic to handle updates and inserts.
  - Clean and transform incoming data from the bronze table.
  - Create a view of the cleaned data.
  - Create an empty silver table.

  - Create a view of the silver table.
  - Use a merge statement to insert new records and update existing records.

Data Modeling (Silver to Gold)

- **Data Modeling**: Decide on the number of tables and the data modeling strategy.
  - Use a star schema with one fact table and multiple dimension tables.
- **Fact Table**: Student performance (student ID, course ID, enrollment date, completion date, time taken).
- **Dimension Tables**:
  - Dim Student (student attributes like name, geography, language).
  - Dim Course (course information like course name, grade level).

Creating Facts and Dimension Tables

1. **Create Lakehouse**: Create a lakehouse for the gold layer (e.g., `LH_gold`).
2. **Notebook**: Use a notebook to create the facts and dimension tables.
3. **Read Silver Data**: Read the data from the silver table.
4. **Create Dimension Tables**: Create the `dim_student` and `dim_course` tables by selecting the appropriate columns from the silver table.
5. **Create Fact Table**: Create the `fact_student_performance` table by selecting the appropriate columns from the silver table.

Data Modeling in Semantic Model

1. **Open SQL Analytics Endpoint**: Open the SQL analytics endpoint for the gold lakehouse.
2. **Create Semantic Model**: Create a new semantic model.
3. **Select Tables**: Choose the fact and dimension tables.
4. **Build Relationships**: Build relationships between the tables.

Report Creation

1. **Open Power BI Desktop**: Open Power BI Desktop.
2. **Connect to Semantic Model**: Connect to the semantic model.
3. **Build Report**: Build a report using the data from the fact and dimension tables.
4. **Publish Report**: Publish the report to the Fabric workspace.

Orchestration

1. **Automate Raw to Landing**: Ensure the ingestion from raw to landing is automated.
2. **Orchestrate Pipelines**: Chain the pipelines together.
   - Raw to landing -> landing to bronze -> silver transform -> gold layer.
3. **Parameterize Notebooks**: Parameterize all notebooks to accept parameters from the pipeline.
4. **Schedule Pipeline**: Schedule the pipeline to run daily.