# Guidance on using Snowflake within the MS Fabric technical stack.

**1. As a Data Source for ELT Pipelines:**

* **Data Factory:** Fabric's Data Factory provides a native **Snowflake connector**. This allows you to:
  + **Copy data** from Snowflake tables into OneLake (in Delta Lake format), a Fabric Data Warehouse, or other supported destinations. You can use the Copy Activity in pipelines for both full and incremental loads.
  + **Use Snowflake as a source in Dataflows Gen2** for visual data transformation before loading into Fabric storage.
  + **Execute Snowflake SQL queries** within a pipeline using the Script Activity for custom data extraction or transformations within Snowflake itself.
  + **Look up data** in Snowflake tables using the Lookup Activity for pipeline control flow logic.
* **Data Engineering (Spark):** You can connect to Snowflake from Fabric Spark notebooks using Spark's JDBC connector. This allows for more complex transformations and data manipulation using Python (PySpark) or Scala, leveraging Snowflake's compute power before bringing data into Fabric.

**2. As a Mirrored Database (Preview):**

* Microsoft Fabric offers a **Mirroring feature** that can directly replicate data from Snowflake into a Fabric mirrored warehouse.
* This provides **near real-time synchronization** of your Snowflake data within Fabric without the need to build and manage complex pipelines.
* The mirrored data can then be queried directly within Fabric using SQL and used for reporting with Power BI in **Direct Lake mode** for high performance.

**3. As a Data Source for Reporting:**

* **Power BI:** You can directly connect Power BI within Fabric to your Snowflake instance using the built-in Snowflake connector. This allows for:
  + **Import mode:** Bringing a snapshot of your Snowflake data into the Power BI semantic model for fast in-memory analysis.
  + **DirectQuery mode:** Querying the live data in Snowflake as you interact with reports, ensuring you always have the latest information.
  + **Direct Lake mode (via Mirroring):** If you've mirrored your Snowflake data into Fabric, Power BI can connect in Direct Lake mode for exceptional query performance directly against the Fabric data.

**Key Considerations for Integration:**

* **Connectivity:** Ensure network connectivity between your Fabric workspace and your Snowflake instance. You might need to configure firewalls or use a gateway in some scenarios.
* **Authentication:** You'll need to configure secure connections using appropriate authentication methods supported by the Snowflake connector (e.g., Snowflake username/password, Microsoft Account).
* **Data Movement Strategy:** Decide whether you want to copy/move data into Fabric for processing and reporting or query Snowflake directly. Mirroring offers a hybrid approach. Consider data volume, performance requirements, and cost implications for each strategy.
* **Transformation Location:** Determine where you want to perform data transformations – within Snowflake using SQL, within Fabric using Dataflows or Spark, or a combination of both. "Transforming as upstream as possible" (in Snowflake) is often a good practice for governance and performance.
* **Cost Optimization:** Be mindful of compute and storage costs on both Snowflake and Microsoft Fabric when designing your integration strategy.

**In essence, Microsoft Fabric is designed to interoperate with existing data platforms like Snowflake. You can leverage Snowflake's robust data warehousing capabilities and seamlessly integrate its data into Fabric for comprehensive ELT processes and powerful reporting with Power BI.**

Sources and related content