**Azure ne işe yarar?**

Azure, Microsoft'un bulut platformudur. Büyük veri analitiği, veri depolama ve dönüştürme için bir dizi hizmet sunar.

**Services:**

1. **Azure Databricks:**Apache Spark tabanlı bir analiz platformudur. Büyük veri analitiğini ve makine öğrenimi süreçlerini kolaylaştırır.
2. **Azure Synapse Analytics:**Büyük veri analitiği ve veri ambarı çözümleri sunar. Veriyi toplama, dönüştürme ve analiz etme süreçlerini tek bir platformda birleştirir.
3. **Azure Power BI:**Veriyi görselleştirmek ve interaktif raporlar oluşturmak için kullanılır.
4. **Azure Data Factory:**ETL (Extract, Transform, Load) ve ELT (Extract, Load, Transform) süreçleri için tasarlanmış bir veri entegrasyon aracı.
5. **Azure Data Lake Storage Gen2:**Yapılandırılmış ve yapılandırılmamış veriler için optimize edilmiş bir depolama çözümü.

**Each will be a simple data ingestion or ELT or ETL tools.**Bu hizmetler, veriyi alma ve dönüştürme işlemlerini kolaylaştırmak için tasarlanmıştır.

**You can also process Data Lake Generation:**Azure, büyük miktarda ham veriyi bir **data lake** içinde saklamanıza olanak tanır.

**We don’t want to process batch and stream data at the same time.**Genellikle toplu (batch) veriler ve akış (streaming) verileri ayrı süreçlerde işlenir.

**We also want to have the data in a data lake.**Verinin tüm formatlardan bağımsız bir şekilde bir **data lake** içinde tutulması, esneklik sağlar.

**With data lakehouse architecture, we have the ability to organize this data efficiently.**Lakehouse mimarisi, bir data lake’in esnekliğini bir veri ambarının (data warehouse) verimliliğiyle birleştirir.

**Büyük veri için analiz tool’u:**Azure'un analiz araçları, büyük veriyi anlamlı hale getirmenize yardımcı olur.

* **Azure Data Factory:** ETL ve ELT işlemleri için.
* **Azure Data Lake Gen2:** Veri depolama.
* **Azure Databricks:** Veriyi işlemek ve dönüştürmek için kompakt bir çözüm.

**Lakehouse mimarisi kurulur:**Veriler üç katmanda düzenlenir:

* **Brown (Ham Veri - Bronze):** Raw, işlenmemiş veri.
* **Silver (İşlenmiş Veri):** Temizlenmiş ve dönüştürülmüş veri.
* **Gold (Analiz İçin Veri):** Analize hazır, optimize edilmiş veri.

**This is more secure compared to a data lake.**Bir data lakehouse, sıradan bir data lake’e göre daha güvenli ve düzenlidir.

**Data Lifecycle:**Veri yaşam döngüsü, verinin üretilmesinden arşivlenmesine veya silinmesine kadar olan süreçleri kapsar. Bu bir “yaşayan operasyon” olarak düşünülebilir.

* **FOA (Focus on Automation):** Veri boru hattı, veri yaşam döngüsünü sürekli hale getiren bir süreçtir. Veriyi bir **data lake**’den alıp dönüştürür, ardından bir sonraki adıma aktarır.

**ETL vs. ELT:**

* **ETL:** Veriler önce dönüştürülür, ardından veri ambarına yüklenir.
* **ELT:** Veriler önce yüklenir, ardından dönüştürme işlemleri yapılır.

**If we have batch and streaming data:**

* Toplu veri için genellikle ETL; akış verileri için gerçek zamanlı işlemler kullanılır.

**When it is like this, it’s not an ETL:**Lakehouse mimarisiyle düzenlenen veri boru hatları genellikle ELT şeklindedir, çünkü ham veriler önce bir data lake'e yüklenir.

**ETL vs. ELT vs. Change Data Capture**

| **Aspect** | **ETL (Extract, Transform, Load)** | **ELT (Extract, Load, Transform)** | **Change Data Capture (CDC)** |
| --- | --- | --- | --- |
| **Workflow Sequence** | Extract → Transform → Load | Extract → Load → Transform | Tracks and applies changes (insert, update, delete) to data. |
| **Primary Use Case** | Structured data transformation and loading into a data warehouse. | Large-scale unstructured data in data lakes or lakehouses. | Keeping systems in sync by tracking changes in source systems. |
| **Data Transformation** | Done before loading into the target system. | Done after loading into the target system. | Only tracks and applies changes, minimal transformation involved. |
| **Processing Style** | Batch | Batch or near real-time | Real-time (incremental updates). |
| **Performance** | Limited by the speed of transformation. | Faster, as transformations occur post-load. | Efficient for incremental updates. |
| **Scalability** | Less scalable for large data. | High scalability for big data workloads. | Scales with system but limited by source system throughput. |
| **Tools/ Technologies** | Azure Data Factory, Informatica, Talend | Azure Data Factory, Azure Synapse, Databricks | Debezium, SQL Server CDC, Azure Data Factory. |
| **Complexity** | Medium (requires preloading transformations). | Low to medium (transformations are deferred). | High (requires monitoring and syncing mechanisms). |
| **Best Suited For** | Legacy systems, structured data workflows. | Big data analytics, data lakes, or lakehouses. | Incremental backups, real-time synchronization. |
| **Limitations** | Time-consuming for large data sets. | Requires powerful compute for post-load transformations. | Requires additional tools for managing and monitoring changes. |

**Change Data Capture (CDC):**

* CDC is useful for real-time synchronization of data between systems or creating incremental backups.
* Requires services that can track changes in source data, such as Azure Data Factory or SQL Server CDC.
* Offers scalable and flexible solutions, avoiding the need to reprocess the entire dataset.

**Azure Workflow for the Project:**

* **Project Deadline:** January 2nd.
* **Azure Environment:**
  + Use **Azure for Students** for resources.
  + Always choose the same **resource group** for consistency.
  + Leverage **real data** for hot path workflows.

**Network Access and Authentication:**

* Ensure **network settings** are unchanged.
* Use **Azure Key Vault** to securely store user credentials and manage encryption (optional but recommended).

**Primary Tools and Steps:**

1. **Azure Data Factory:**
   * Create and manage data pipelines.
   * Go to "Factory Resources" to configure workflows.
   * Use **Pipeline Expression Builder** for schema definitions and data processing logic.
2. **Azure Databricks:**
   * Build clusters and perform transformations on data.
   * Combine services for seamless workflows.
   * Implement **Lakehouse Architecture**:
     + Bronze (raw data) → Silver (analyzable format, e.g., Parquet) → Gold (final, relational format).
3. **Relational Databases (SQL):**
   * Use SQL for queries and connections.
   * **CSV/JSON Formats:** For simplicity, load raw data directly without cleaning.

**Key Concepts for Implementation:**

1. **Data Pipeline Workflow:**
   * Sequentially link services like Data Factory and Databricks to enable end-to-end processing.
   * Use **Linked Services** in Azure Data Factory to connect to SQL or other sources.
2. **Lakehouse and Node Management:**
   * Use HDFS (Hadoop Distributed File System) environment.
   * Organize nodes (including master and backup nodes) for efficient management.
3. **Clustering:**
   * Create clusters for scalable processing.
   * Follow the "n-over-2" rule for cluster setup.
4. **Parquet Format:**
   * For intermediate Silver layer storage, use optimized formats like **Parquet** to improve analytics performance.

**Final Exam Tips:**

* Focus on pipeline creation, connecting services, and designing workflows.
* Understand Lakehouse architecture (Bronze, Silver, Gold layers).
* Be prepared to work with authentication (Azure Key Vault) and schema definitions.

Databricks Services

**Databricks Services** help manage and process large datasets efficiently. They allow you to:

1. Build scalable **data pipelines** for batch and streaming data.
2. Perform **data transformations** (e.g., cleaning, aggregating) on big data.
3. Use distributed clusters for parallel processing.
4. Leverage **machine learning** and **analytics workflows** on large datasets.

Parquet Format and Delta Lake

* **Parquet Format:**
  + A columnar storage format optimized for analytics workloads.
  + Efficient for compression and faster query performance, particularly with big data.
* **Delta Lake:**
  + Built on **Parquet**, but includes additional features such as:
    - **ACID transactions** for reliability.
    - **Versioning:** Enables time travel and auditing of table changes.
    - **Streaming and Batch Support:** Seamlessly handles real-time data alongside batch data.

In essence, **Delta** = Enhanced **Parquet** with transactional capabilities.

Lakehouse Workflow

The **Lakehouse Architecture** bridges the gap between unstructured data in a data lake and structured data in a warehouse.

1. **New Table Format (Hot Data):**
   * Example: **Customer Dimension Table**
   * When the table format changes or new hot data is generated, it is stored in **Delta Lake** format within the **Data Lakehouse**.
2. **Bronze → Silver → Gold Layers:**
   * **Bronze Layer:** Raw, unprocessed data (Data Lake format).
   * **Silver Layer:** Cleaned and structured for intermediate analysis (e.g., Parquet format).
   * **Gold Layer:** Fully processed, optimized data for reporting and business use.

Hot Data Workflow in Databricks

* **Hot Data:**
  + Frequently accessed or recently updated data.
  + Stored in Delta format for immediate analysis.
* **Integration with Data Lake Format:**
  + Hot data is pushed into a Data Lake in **Delta** or **Parquet** format, depending on the processing stage.

This system ensures that both raw and processed data are stored efficiently while remaining accessible for real-time analytics.