**SUMMARY-DAY14**

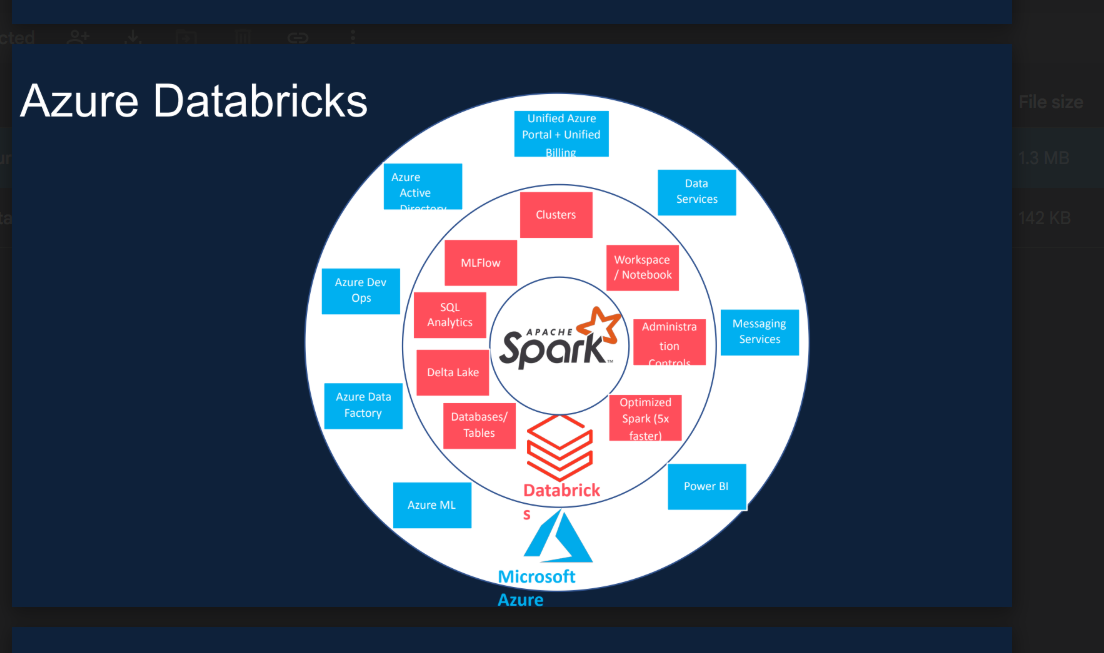
**Name:**Tejaswini Gokanakonda

**Roll no:**DE142

**Date:**27-11-2024

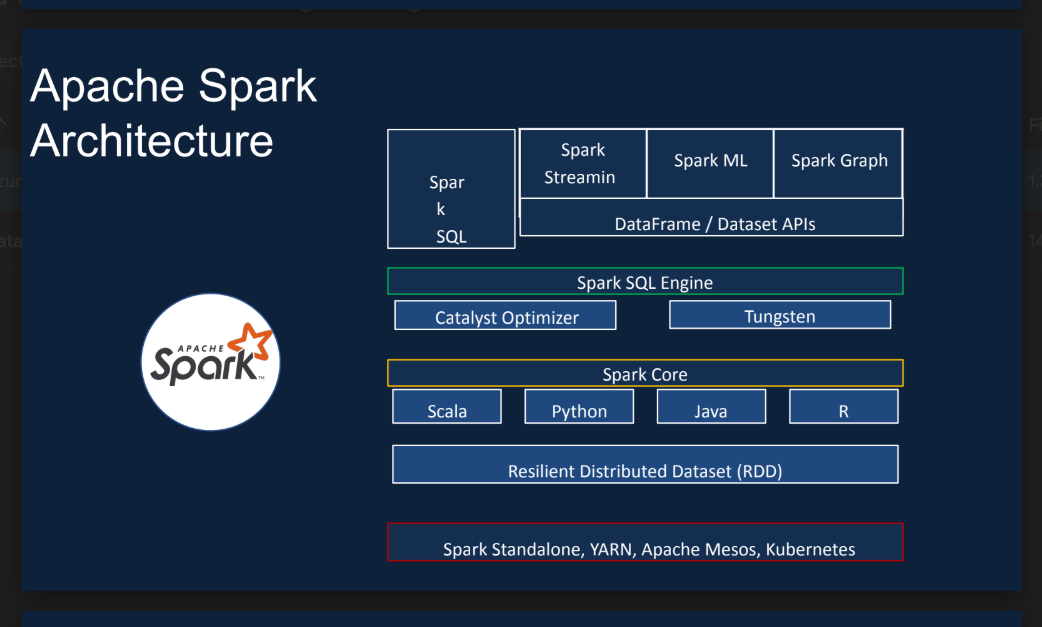
**Azure Databricks**

* **Azure Databricks**: A cloud-based platform combining Apache Spark and Azure services.
* **Key Features**:
  + Built on **Apache Spark**, offering high-speed distributed computing.
  + Unified analytics for **SQL, Streaming, Machine Learning (ML)**, and **Graph Processing**.
  + Integrated with Azure services like **Azure Data Lake, Azure ML, Power BI, Azure SQL Database**.
  + **Optimized Spark (5x faster)** for superior performance.
  + Collaboration tools: Notebooks, Clusters, Jobs, and Workspaces.
  + Unified billing via **Azure Portal** with **Azure Active Directory** integration.



**Apache Spark Architecture**

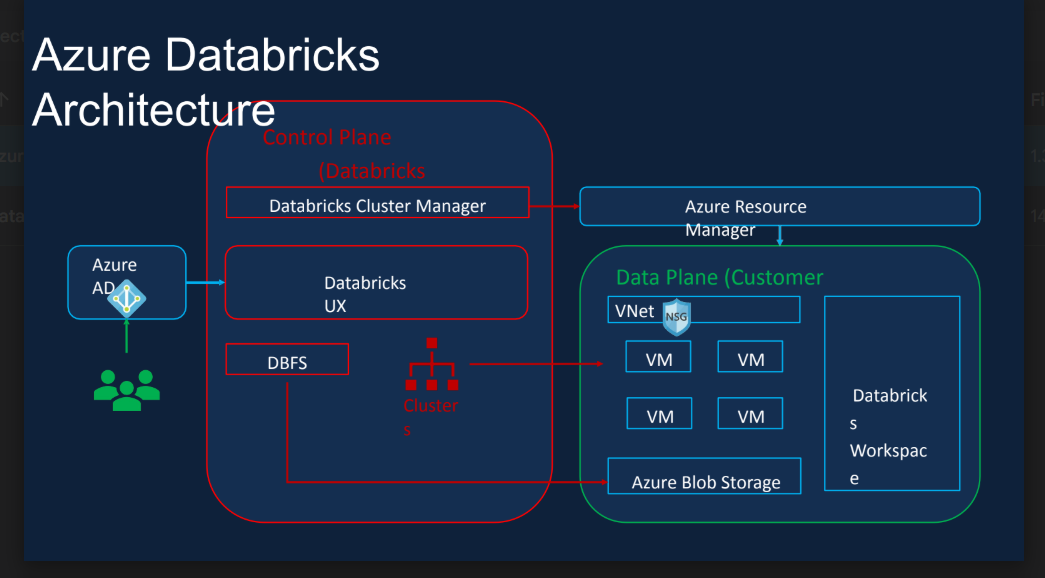
* **Core Components**:
  + **Spark Core**: Task scheduling, in-memory computation, and fault tolerance.
  + **Spark SQL**: SQL-like interface for structured data analysis.
  + **Spark Streaming**: Real-time data streaming and processing.
  + **Spark MLlib**: Tools for scalable machine learning.
  + **Spark GraphX**: For graph processing and analysis.



* **APIs**: Supports **Scala, Python, Java, R**.
* **Execution Modes**:
  + Spark Standalone, YARN, Apache Mesos, Kubernetes.
* **Key Features**:
  + Resilient Distributed Dataset (RDD): Immutable distributed data objects.
  + Catalyst Optimizer & Tungsten: Enhances SQL query performance.

**Azure Databricks Architecture**

* **Control Plane**:
  + Manages metadata and user interface via Databricks subscription.
* **Data Plane**:
  + Executes workloads using customer subscriptions.
  + Uses Azure Virtual Network (VNet) and virtual machines for data security.
* **Components**:
  + **Notebooks**: Collaborative workspace for code and analysis.
  + **Clusters**: Execution environment for processing workloads.
  + **Jobs**: Task automation for running applications.
  + **Data Models**: Integrated data management for seamless analytics.



**Databricks Clusters**

* **Cluster Types**:
  + **All-Purpose Clusters**: Persistent, shared by multiple users, suitable for interactive workloads.
  + **Job Clusters**: Temporary clusters created per job, terminated after use, cost-efficient.
* **Configuration Options**:
  + **Single Node**: For simple workloads.
  + **Multi-Node**: For distributed processing.
  + **Access Modes**:
    - **Shared**: Multiple users, collaborative.
    - **Custom**: Legacy or specialized configurations.
  + **Features**:
    - **Auto Scaling**: Dynamically adjusts the number of worker nodes.
    - **Auto Termination**: Shuts down clusters after inactivity (default: 120 minutes).

**Databricks Pricing**

* **Databricks Unit (DBU)**: Measure of processing power.
* **Pricing Factors**:
  + Workload type (All-Purpose, Jobs, SQL, Photon).
  + Tier (Standard or Premium).
  + VM type (General Purpose, GPU, Optimized).
  + Purchase Plan (Pay-As-You-Go or Pre-Purchase).
* **Example Cost**:
  + Small single-node cluster (Premium tier): **$0.76/hour**.
  + Typical student workload: $15–$25 for 20–30 hours (Pay-As-You-Go).

**Databases vs. Data Warehouses vs. Data Lakes vs. Delta Lake**

**1. Database**

* **Definition**: Organized collection of data supporting **Online Transaction Processing (OLTP)**.
* **Features**:
  + Supports structured and semi-structured data.
  + Optimized for real-time read/write operations.
  + Ensures data integrity with **ACID transactions**.
  + Includes indexing and query languages (e.g., SQL).
* **Types**:
  + **Relational**: MySQL, PostgreSQL, Oracle.
  + **Non-Relational**: MongoDB, Redis, Cassandra.
* **Use Cases**: Patient records, online store items, IoT data.

**2. Data Warehouse**

* **Definition**: Centralized system for structured data, optimized for **Online Analytical Processing (OLAP)**.
* **Features**:
  + Stores historical and current data from multiple sources.
  + Uses **ETL (Extract, Transform, Load)** processes.
  + Pre-defined schema; integrates with **BI tools** (e.g., Power BI, Tableau).
* **Examples**: Snowflake, Amazon Redshift, Azure Synapse, Google BigQuery.
* **Use Cases**: Quarterly reports, business forecasting, trend analysis.

**3. Data Lake**

* **Definition**: Repository for raw, unprocessed data from disparate sources.
* **Features**:
  + Schema-on-read flexibility: Ingests structured, semi-structured, and unstructured data.
  + Supports formats like JSON, Parquet, CSV, and Avro.
  + Ideal for machine learning and predictive analytics.
* **Examples**: AWS S3, Azure Data Lake Gen2, Google Cloud Storage.
* **Use Cases**: Storing multimedia files, logs, raw datasets for future analysis.

**4. Delta Lake**

* **Definition**: A storage layer on top of data lakes that adds **reliability, performance**, and **ACID transactions**.
* **Advantages**:
  + Handles streaming and batch processing workloads.
  + Provides schema enforcement, data versioning, and improved query performance.

**Comparison Table**

| **Feature** | **Database** | **Data Warehouse** | **Data Lake** |
| --- | --- | --- | --- |
| **Primary Use** | Operational (OLTP) | Analytical (OLAP) | Analytical/Storage |
| **Data Type** | Structured/Semi-structured | Structured/Semi-structured | Structured/Semi/Unstructured |
| **Schema** | Rigid/Flexible | Pre-defined | Schema-on-read |
| **Freshness** | Real-time | Periodic (ETL-dependent) | Periodic (ETL-dependent) |
| **Users** | Developers | Business Analysts/Data Scientists | Analysts, Data Engineers |
| **Examples** | MySQL, MongoDB | Snowflake, Redshift | Azure Data Lake Gen2, AWS S3 |

**When to Use?**

* **Database**: For powering day-to-day transactional applications.
* **Data Warehouse**: For large-scale historical and structured data analysis.
* **Data Lake**: For flexible storage of raw data with machine learning use cases.
* **Delta Lake**: To add reliability and performance to existing data lakes.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |