**Day 17 Notes**

**Name : Podutur Lahari - DE126**

**Date : 02-11-2024**

#### **Topics Covered:**

1. Scheduling Workflow Jobs:
   * Configuring and running jobs on Azure Databricks.
2. Alert Configurations:
   * Notifications through email, webhook, or Slack.
3. Submitting Jobs Using Job Clusters:
   * Running jobs in Databricks Job Clusters.
4. Creating Jobs on Databricks Using Notebooks:
   * Use the Schedule button to create and manage scheduled notebook jobs.
5. Hands-on Tasks:
   * Running jobs on ingestion load tutorial cluster.
   * Configuring parameters for notebooks using Delta or PySpark programs.
   * Scheduling notebooks as jobs.

#### **Key Concepts:**

1. Databricks Jobs:
   * Definition: Primary unit for scheduling and orchestrating workloads.
   * Composition: Consists of one or more tasks.
   * Capabilities:
     + Execute custom logic (e.g., Spark, SQL, Python, ML).
     + Configure compute resources.
     + Set schedules and triggers.
     + Implement conditional logic for task relationships.
   * Approach:
     + Procedural (e.g., control flow between tasks).
     + Declarative (e.g., Delta Live Tables pipelines).
2. **Minimum Job Configuration:**
   * Source code (e.g., notebooks).
   * Compute resources (serverless, classic jobs compute, or all-purpose compute).
   * Schedule or manual trigger.
   * Unique name.
3. **Tasks in Jobs:**
   * Represent individual logic steps (e.g., notebooks, SQL, JAR files, DLT pipelines).
   * Execution order controlled through dependencies.
   * Share context using task values.
4. **Control Flow Options:**
   * Triggers, retries, conditional tasks (If/Else, For Each), duration thresholds, concurrency.

#### **Trigger Types:**

1. Scheduled: Automatically based on set intervals.
2. File Arrival: Triggered when a file is detected.
3. Continuous: For real-time processing.
4. Manual:
   * Rarely used for migrations or infrequent workloads.
   * Validated with external orchestration tools or manual interventions.

#### 

#### 

#### **Infrastructure-as-Code (IaC) Approach:**

* Use Databricks Asset Bundles (DABs):
  + YAML definitions for jobs.
  + Managed via Databricks CLI.
  + Shareable across environments (development, staging, production).

#### **Key Concepts of Azure Data Lake Storage (ADLS):**

1. **Blob Storage:**
   * Stores unstructured data in a flat namespace.
   * Organizes blobs using "/" characters for virtual "folders."
   * Access via HTTP or HTTPS.
2. **ADLS Gen2:**
   * Built on blob storage; adds a hierarchical namespace:
     + Organizes blobs into directories.
     + Supports metadata for each directory and file.
     + Performs atomic operations (e.g., directory renames and deletes).
   * Improves performance for analytical workloads.
   * Reduces costs for data analysis compared to flat namespaces.
3. **Hierarchy Configuration:**
   * Disabled:
     + Acts as standard Azure Blob storage.
     + Suitable for data archiving or storing web assets (e.g., images, media).
   * Enabled:
     + Optimized for analytics; integrates Blob and ADLS Gen2 APIs.

#### 

#### 

#### **Stages of Big Data Processing:**

1. **Ingest:**
   * Acquires source data from files, logs, and unstructured formats.
   * Technology choice depends on data transfer frequency:
     + Batch Ingestion: Azure Synapse Analytics or Azure Data Factory.
     + Real-time Ingestion: Apache Kafka (HDInsight) or Stream Analytics.
2. **Store:**
   * Places ingested data in scalable and secure solutions like Azure Data Lake Storage Gen2.
   * Compatible with common big data processing tools.
3. **Prep and Train:**
   * Prepares data and trains ML models.
   * Common technologies:
     + Azure Synapse Analytics
     + Azure Databricks
     + Azure HDInsight
     + Azure Machine Learning
4. **Model and Serve:**
   * Presents processed data to users.
   * Technologies include:
     + Visualization tools (e.g., Microsoft Power BI).
     + Analytical data stores (e.g., Azure Synapse Analytics).
   * Combines multiple tools based on business needs.

#### **Tips:**

* Use Blob Storage for simple storage needs (no analysis).
* Use ADLS Gen2 for analytics-focused solutions with hierarchical namespace enabled.