Build a Spark Streaming Pipeline with Synapse and CosmosDB

Overview
Azure Cosmos DB is a globally distributed, multi-model database service provided by Microsoft Azure. It is designed to handle massive amounts of data and deliver low-latency access to applications around the world. Cosmos DB offers several key features and capabilities that make it a popular choice for building scalable and highly available applications.

Here are some important aspects of Cosmos DB:

- **Global distribution:** Cosmos DB allows you to distribute your data across multiple Azure regions, enabling low-latency access to users worldwide. It ensures data replication and availability across different regions, providing high availability and disaster recovery options.
- **Multi-model:** Cosmos DB supports multiple data models, including document, key-value, graph, column-family, and table. This flexibility allows developers to choose the most appropriate model for their application's needs.
- **Scalability:** Cosmos DB offers horizontal scaling, allowing you to elastically scale throughput and storage as your application demands increase. It automatically handles the distribution of data across partitions and enables seamless scaling without any downtime.
- **Low latency:** Cosmos DB provides single-digit millisecond latency for both reads and writes globally. This fast response time makes it suitable for real-time applications and scenarios that require low-latency data access.
- **Multi-API support:** Cosmos DB supports various APIs, including SQL (DocumentDB), MongoDB, Cassandra, Gremlin (graph), and Azure Table Storage. This compatibility allows developers to leverage their existing skills and use familiar programming models when working with Cosmos DB.
- **Consistency models:** Cosmos DB offers a choice of five well-defined consistency models: strong, bounded staleness, session, consistent prefix, and eventual consistency. This allows developers to select the desired level of consistency based on their application requirements.

Aim:

The objective of this project is to construct a spark streaming pipeline, utilizing the capabilities of Azure Synapse Analytics and Azure Cosmos DB. The pipeline will incorporate the implementation of window functions, specifically focusing on two types: tumbling window functions and sliding window functions. These window functions play a crucial role in data processing and analytics by facilitating calculations on specific subsets of data. Additionally, the project will involve working with joins to combine relevant data from different sources. Furthermore, the project will explore the creation of logic apps, enabling the configuration of email alerts for specific events or conditions. By encompassing these components, the project aims to showcase the integration of Azure

Synapse Analytics and Azure Cosmos DB, as well as the utilization of window functions, joins, and logic apps for comprehensive data analysis and processing.

Tech Stack
➔
Language: Python, SQL
➔
Package: PySpark
➔
Services: Azure Blob Storage (ADLS Gen2), Azure Synapse Analytics, Logic Apps, Azure Cosmos DB

Window functions:

Window functions, also known as analytic functions, are a powerful feature in SQL that allows you to perform calculations across a set of rows within a query result. Tumbling window functions and sliding window functions are two types of window functions used in data processing and analytics. They are used to define and operate on specific subsets of data within a larger dataset.

- **Tumbling Window Functions:** A tumbling window function divides the dataset into non-overlapping, fixed-size windows. Each window includes a specified number of rows or a specific time range. Tumbling windows "tumble" or roll over the dataset without any overlap. Tumbling windows are useful for performing calculations on distinct and separate subsets of data. For example, you can calculate the sum of sales for each day using a tumbling window of 24 hours.
- **Sliding Window Functions:** A sliding window function, on the other hand, creates overlapping windows as it moves through the dataset. The window slides across the dataset, including a specified number of preceding or following rows or a specific time range. Sliding windows enable computations that consider recent or historical data points together. For example, you can calculate a moving average of sales by using a sliding window that includes the previous seven days.

Key Takeaways:
- Understanding the use of NoSQL Databases
- Difference between Spark Streaming and Spark Batch Processing
- Creating Azure Cosmos DB instance
- Utilizing Azure Synapse Analytics and Azure Cosmos DB to construct a spark streaming pipeline
- Understanding the need for Window functions
- Understanding the Window functions in depth
- Understanding different types of Window functions
- Implementing tumbling window functions

- Implementing sliding window functions
- Creating containers in Cosmos DB
- Inserting JSON object in containers of Cosmos DB
- Integrating Cosmos DB in Azure Synapse Analytics
- Creating Logic Apps for email alerts.

Architecture diagram: