1. **Problem Statement**

Title: Real-time Stream Processing of Bank Data using Apache Kafka on Azure

Description:

The project aims to establish a real-time data streaming pipeline for processing a bulk dataset containing bank data. This data needs to be ingested, transformed into a stream, and efficiently transported using Apache Kafka. The primary objectives are to set up an Azure virtual machine instance, configure Apache Kafka, and develop a Python-based producer to stream each record from the bank dataset to a Kafka topic. Finally, a Kafka consumer will retrieve and process these records in real-time.

1. **Solution Architecture**

2.1) Azure Virtual Machine Instance:

Create a virtual machine instance in Azure to host the Kafka infrastructure.

Install and configure the necessary dependencies and tools on the virtual machine.

2.2) Apache Kafka Setup:

Deploy and configure Apache ZooKeeper, which is essential for Kafka coordination.

Install and configure the Kafka server, establishing topics and partitioning as needed.

Configure the Kafka producer and consumer for data ingestion and retrieval.

2.3) Data Ingestion and Streaming:

Utilize Python and Pandas to convert the bulk bank dataset into a stream of records.

Develop a Kafka producer script to send each record to the Kafka topic.

2.4) Real-time Data Processing:

Set up a Kafka consumer application to subscribe to the Kafka topic.

Implement data processing logic within the consumer to perform real-time analysis or storage of bank data.

1. **Workflow**

3.1) Azure Virtual Machine Setup:

Provision a virtual machine instance in Azure.

Configure the virtual machine with the necessary resources, including storage and network settings.

Install and configure the required software, libraries, and tools for Kafka deployment.

3.2) Apache Kafka Deployment:

Deploy Apache ZooKeeper for coordination and Kafka for data streaming on the virtual machine.

Configure Kafka topics and partitions based on data processing requirements.

3.3) Data Preparation and Ingestion:

Acquire the bulk bank dataset.

Use Python and Pandas to transform the dataset into a stream of records.

Develop a Kafka producer script to push each record to the appropriate Kafka topic.

3.4) Real-time Data Processing:

Implement a Kafka consumer application that subscribes to the Kafka topic.

Process each incoming record in real-time using custom logic.

Perform actions such as storing data in a database, triggering alerts, or generating real-time reports as needed.

3.5) Monitoring and Maintenance:

Set up monitoring and logging to ensure the health and performance of the Kafka infrastructure.

Schedule routine maintenance tasks, including backups and updates, to keep the system running smoothly.

By following this workflow and solution architecture, the project aims to achieve the real-time processing of bank data, enabling timely insights and actions based on the streaming data.