Cloud project

Lab 3: Kafka connects for Redis and MySQL

Objective:

- Deploy Tabular and key-Value data storage to GKE.
- Get familiar with Key-Value data storage
- Get familiar with Kafka Connectors and their configuration.
- Configure and use Kafka source connector to Redis.
- Configure and user MySQL sink and source Kafka connectors.

Lab3 repository: https://github.com/goergedaoud/SOFE4630U-tut3.git

(Links to an external site.)

Procedure:

1. **Watch the first three videos for Kafka connectors (focus on the concepts, not the details) from https://www.confluent.io/blog/kafka-connect-tutorial/**

Describe the following:

- **Sink and Source connectors**
  - A Source Connector is responsible for bringing data into Kafka (with the aid of Source Tasks). This could be a database, stream tables, and even message brokers. Furthermore, the source connector can collect metrics from application servers into Kafka topics, which allows the data available for stream processing with low latency.
  - Sink Connector is responsible for getting data out of Kafka (with the help of Sink Tasks) A sink connector provides data from Kafka topics to other systems which could be Elasticsearch, batch systems like Hadoop, or any database.

- **The applications/advantages of using Kafka Connectors with data storage.**
  - Data-Centric Pipeline - Connect pulls or pushes data to Kafka using meaningful data abstractions.
  - Connect can be used with both streaming and batch-oriented systems on a single node (standalone) or scaled to a company-wide service (distributed).

- Reusability and Extensibility - Connect makes use of existing connectors or extends them to meet your specific requirements, resulting in a faster time to market.
- The main advantage of using Kafka connectors is that they are flexible and modular.
- The pipeline allows for connecting meaningful data abstractions to pull and push data to Kafka. The connection runs stream and batch systems which can scale to organization-wide service. Connections extend them to tailor the user's needs along with using existing connectors.

- **How do Kafka connectors maintain availability?**
  - The connector interfaces with outside technology and mediates the data into Kafka connect.
  - Ingests whole databases and feeds table updates to Kafka topics using the source connector. A source connector can also collect metrics from all of your application servers and store them in Kafka topics, allowing for low-latency stream processing.
  - Furthermore, Kafka connectors also have task rebalancing and fault tolerance for Kafka connect. If a worker fails unexpectedly, every other worker will detect it automatically and redistribute connectors and tasks across available workers.

- **List the popular Kafka converters for values and the properties/advantages of each.**
  - On the route into or out of Kafka, converters serialize or deserialize the data.
  - Avro: Sterilizes record keys and values, it is very compact and efficient.
  - Checks to make sure every record has a proper structure
  - Protobuf: Ensures signals don't get lost between other apps, it processes
  - information very quickly
  - String: able to define conversion between strings and objects and control
  - their behavior.
  - JSON: defines which JSON converter is used to convert an object.
  - ByteArray: returns an array of bytes
  - Kafka converters take in a specific record set from different types of data schemes and use a converter to convert the data to format it into different types for data ingestion.
  - There are several guidelines for choosing serialization formats such as

- Schema (ie. provides a contract between services) Message formats such as Avro and Protobuf have strong schema support. JSON and delimited strings do not have much schema support.
- Ecosystem compatibility (Avro, JSON, and Protobuf typically first-class citizens on Confluent Platform.
- Message size (JSON relies on compression where Avro and Protobuf are binary formats and have a smaller message size
- Language support (Avro is strongly Java-based and with Protobuf, Go is typically more used.

3. Search the internet to answer the following question:

- **What's a Key-Value (KV) database?**
  - It uses a key-value method to store data, it can store, retrieve, and manage arrays of data, essentially its a hash table
  - A key-value database is a nonrelational database that stores data using a simple key-value mechanism. Data is stored in a key-value database as a collection of key-value pairs, with a key serving as a unique identifier. Both keys and values can be any type of object, from simple to sophisticated compound objects.
- **What are KV databases' advantages and disadvantages?**
  - Advantages
    - Scalability: scalable because its ability to take a lot of requests
    - Speed: able to process constant requests for reading and writes
    - Flexibility
  - Disadvantages
    - Only optimized for data with a single key and value
    - Not very well made for reading/looking up data
  - A key-value database is a nonrelational database that stores data using a simple key-value mechanism. Data is stored in a key-value database as a collection of key-value pairs, with a key serving as a unique identifier. Both keys and values can be any type of object, from simple to sophisticated compound objects.

- **List some popular KV databases**
  - Amazon DynamoDB
  - Amazon ElastiCache
  - Redis
  - Couchbase

- ScyllaDB
- Aerospike
- Hbase
- InterSystems IRIS

4. **Follow the following videos to deploy and use Redis and MySQL databases using GKE.**
5. Video Link 1 https://drive.google.com/file/d/1xKWV5tuUGVSzRFFYwpcoMC4Ej1bHWmVN /view ?usp=sharing Video
6. Link 2 https://drive.google.com/file/d/1CX2R8Fy-pRHCvH52WXeZz-biDvJIppu5/vie w?usp =sharing
7. Video Link 3 https://drive.google.com/file/d/1LIymU5CsAxJHDhBAnI4Kbvx2crXGYjcp/view ?us p=sharing
8. Video Link 4 https://drive.google.com/file/d/1Z9zVGNKGTgdBEkj99v_35YwMak9fwUla/vie w?us p=sharing
7. as your dataset.

Record a video showing the configuration of Kafka connectors, producers' python script, a proof of successfully stored data into data storage.

9. List some possible applications that can be implemented by using the uploaded dataset.

IT applies to a variety of AI applications, machine learning. Practical uses could include object detection or medical databases. By storing the data on the cloud, it could privately be authorized that anyone from anywhere can access it and have unlimited services.