Cloud Computing
Project Deliverable:
Data Storage Implementation

March 18, 2022

Group 7
Owais Quadri 100697281
Tegveer Singh 100730432
Danial Asghar 100671850
Shayan Sepasdar 100722542

GitHub Link
https://github.com/danialasghar/Cloud-Group-7

Questions

Sink and Source connectors:
Sink: Data in Kafka in form of Processed Strings and Application producing data for a topic that needs to be stored somewhere else
Source: Relational Databases, Message Queues, etc that need to be ingested into the Kafka System

The applications/advantages of using Kafka Connectors with Data storage
Manage interaction between Kafka Connect system and any external technology that might send/fetch data to and from the Kafka System. Most important and highly configurable part of the Kafka pipeline. Acts as an interface between a source and sink by passing compatible data.

How do Kafka connectors maintain availability
Moving data in and out of popular data storage systems, local or cloud can be used as backups to store data for Kafka. Different connectors attached to the same kafka pipeline are independent of each other and provide loose coupling which increases availability of the system. Along with this, convertors and transforms also help in avoiding other threats to availability.

List the popular Kafka convertors for values and properties/advantages of each
All convertors require a key value convertor property
The following are the commonly used Kafka Convertors:

1. JSON
    - Conversion from JSON schema to Kafka connect schema
    - Simple and organized
2. Avro:
    - Conversion from Avro Schema to kafka connect schema and vice versa
    - Login credentials and SSL can be provided
    - Smaller in size than JSON
3. Protobuf
    - Much faster than Avro at serializing data

Whats a Key-Value(KV) database
This database stores data as a collection of key-value pairs where the key acts as a unique identifier for a value.

List the advantages and disadvantages of KV database
Advantages:

- Simple data format leading to quick read and write operations
- No restrictions on the type of keys and values

Disadvantages:

- Lookup is of linear order
- Limited to single key and value pairs

List some popular KV databases

NoSQL databases like DynamoDB and MongoDB are common examples of popularly used key-value databases.

List some applications that can be implemented using the uploaded dataset

The NCLT dataset includes imagery and sensor data for lidar, planar lidar, GPS and proprioceptive sensors. It was created by the University of Michigan to provide research focused on long-term autonomous operation in changing environments. The dataset can be used for object recognition as it contains a lot of different types of objects like people, bikes, cars, etc. Along with just object recognition, this could be applied in technologies for autonomous vehicles.

Videos

This video shows the execution process of setting up a MySQL Source on Confluent cloud:
https://drive.google.com/file/d/116uyWEkZuunDPD5DsFv6379TbW2Mb0dV/view?usp=sharing

This video shows the process of setting up a MySQL Sink on Confluent cloud:
https://drive.google.com/file/d/1XgNqKh-0DR0yYeKMNWo9-ZzrrU5LyX8S/view?usp=sharing

This video shows the process of setting up a Redis Sink on Confluent cloud:
https://drive.google.com/file/d/1lE5FtxndGVqGO5CyVL15Ltq83FfI2-qh/view?usp=sharing

Question 7: This video shows sending csv data from a file to the Kafka cluster and storing it in MySQL. The provided dataset was not used because it was too large of a file and we didn't want to get charged for sending so many messages:
https://drive.google.com/file/d/1E9Tf4dhnUTSRNROzGihXo7N31S_dAWMP/view?usp=sharing

Question 7: This video shows sending multiple images as data to the Kafka cluster and storing it in Redis, with their name defining the Key. The provided dataset was not used because it was 100G+ and we didn't want to get charged for sending so many messages:
https://drive.google.com/file/d/1EgkWkXZOEifeUQfHEUw2n92x0f9h6DTd/view?usp=sharing

Screenshots

Setup and configure GKE

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl apply -f mysql-pvc.yaml
persistentvolumeclaim/mysql-volumeclaim created
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl apply -f mysql-app.yaml
service/mysql created
deployment.apps/mysql created
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
mysql-7dcb5fd764-jlvbz   1/1     Running   0          84s
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$
```

Deploy and use MySQL on GKE

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl exec -it mysql-7dcb5fd764-jlvbz -- mysql -uuser -pSOFE4630U
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use myDB;
Database changed
mysql> select * from myDB.test;
ERROR 1146 (42S02): Table 'myDB.test' doesn't exist
mysql> exit;
Bye
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$
```

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl exec -it mysql-7dcb5fd764-jlvbz -- mysql -uuser -pSOFE4630U<<<"select * from myDB.test;"
Unable to use a TTY - input is not a terminal or the right kind of file
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 1146 (42S02) at line 1: Table 'myDB.test' doesn't exist
command terminated with exit code 1
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$
```

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl exec -it mysql-7dcb5fd764-jlvbz -- mysql -uuser -pSOFE4630U<<<"select * from myDB.test;"
Unable to use a TTY - input is not a terminal or the right kind of file
mysql: [Warning] Using a password on the command line interface can be insecure.
id      name    email   department      modified
1       alice   alice@abc.com   eng.    2022-03-18 03:22:14
2       bob1    bob1@abc.com    sales   2022-03-18 03:22:14
3       bob2    bob2@abc.com    sales   2022-03-18 03:22:14
4       bob3    bob3@abc.com    sales   2022-03-18 03:22:14
5       bob4    bob4@abc.com    sales   2022-03-18 03:22:14
6       bob5    bob5@abc.com    sales   2022-03-18 03:22:14
7       bob6    bob6@abc.com    sales   2022-03-18 03:22:14
8       bob7    bob7@abc.com    sales   2022-03-18 03:22:14
9       bob8    bob8@abc.com    sales   2022-03-18 03:22:14
10      bob9    bob9@abc.com    sales   2022-03-18 03:22:14
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$
```

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ mysql -uuser -pSOFE4630U -h35.226.72.125
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT * FROM myDB.test;
+----+-------+---------------+-------------+---------------------+
| id | name  | email         | department  | modified            |
+----+-------+---------------+-------------+---------------------+
|  1 | alice | alice@abc.com | eng.        | 2022-03-18 03:22:14 |
|  2 | bob1  | bob1@abc.com  | sales       | 2022-03-18 03:22:14 |
|  3 | bob2  | bob2@abc.com  | sales       | 2022-03-18 03:22:14 |
|  4 | bob3  | bob3@abc.com  | sales       | 2022-03-18 03:22:14 |
|  5 | bob4  | bob4@abc.com  | sales       | 2022-03-18 03:22:14 |
|  6 | bob5  | bob5@abc.com  | sales       | 2022-03-18 03:22:14 |
|  7 | bob6  | bob6@abc.com  | sales       | 2022-03-18 03:22:14 |
|  8 | bob7  | bob7@abc.com  | sales       | 2022-03-18 03:22:14 |
|  9 | bob8  | bob8@abc.com  | sales       | 2022-03-18 03:22:14 |
| 10 | bob9  | bob9@abc.com  | sales       | 2022-03-18 03:22:14 |
+----+-------+---------------+-------------+---------------------+
10 rows in set (0.04 sec)

mysql>
```

```
mysql> drop table myDB.test;
Query OK, 0 rows affected (0.09 sec)

mysql> SELECT * FROM myDB.test;
ERROR 1146 (42S02): Table 'myDB.test' doesn't exist
mysql> exit;
Bye
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ mysql -uuser -pSOFE4630U -h35.226.72.125 < sc1.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ mysql -uuser -pSOFE4630U -h35.226.72.125
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT * FROM myDB.test;
+----+-------+---------------+------------+---------------------+
| id | name  | email         | department | modified            |
+----+-------+---------------+------------+---------------------+
|  1 | alice | alice@abc.com | eng.       | 2022-03-18 03:43:26 |
|  2 | bob1  | bob1@abc.com  | sales      | 2022-03-18 03:43:26 |
|  3 | bob2  | bob2@abc.com  | sales      | 2022-03-18 03:43:26 |
```

Deploy and use Redis on GKE

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl apply -f redis-pvc.yaml
persistentvolumeclaim/redis-volumeclaim created
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl apply -f redis-app.yaml
service/redis created
deployment.apps/redis created
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl get pods
NAME                     READY   STATUS              RESTARTS   AGE
mysql-7dcb5fd764-jlvbz   1/1     Running             0          142m
redis-56644c686c-btqpq   0/1     ContainerCreating   0          7s
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl get service
NAME         TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)          AGE
kubernetes   ClusterIP      10.12.0.1      <none>           443/TCP          3h18m
mysql        LoadBalancer   10.12.15.91    35.226.72.125    3306:30070/TCP   143m
redis        LoadBalancer   10.12.3.218    34.122.150.120   6379:31187/TCP   63s
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$
```

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ kubectl exec -it redis-56644c686c-btqpq -- redis-cli
127.0.0.1:6379> auth SOFE4630U
OK
127.0.0.1:6379> set k1 test
OK
127.0.0.1:6379> get k1
"test"
127.0.0.1:6379> set key1 98.2%
OK
127.0.0.1:6379> keys *
1) "key1"
2) "k1"
127.0.0.1:6379>
```

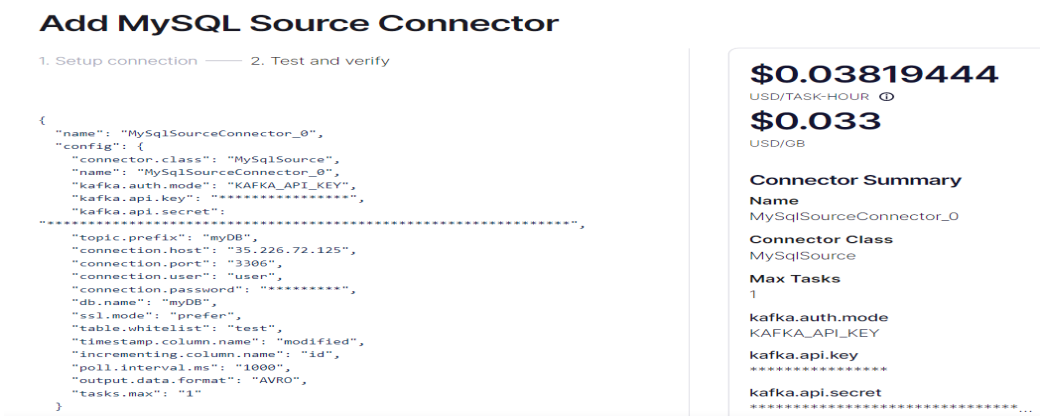
```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ redis-cli -h 34.122.150.120
34.122.150.120:6379> keys *
(error) NOAUTH Authentication required.
34.122.150.120:6379> auth SOFE4630U
OK
34.122.150.120:6379> keys *
1) "Course"
2) "key1"
3) "k1"
34.122.150.120:6379>
```

Execute redis_access.py

```python
import redis        # pip install redis
ip="34.122.150.120"
r = redis.Redis(host=ip, port=6379, db=0,password='SOFE4630U')
v=r.get('key1');
print(v);
r.set('key1','30'.encode('utf-8'));
```

```
tegveer2211@cloudshell:~/SOFE4630U-tut3/GKE (cloud-milestone-3)$ redis-cli -h 34.122.150.120
34.122.150.120:6379> auth SOFE4630U
OK
34.122.150.120:6379> get key1
"30"
34.122.150.120:6379>
```

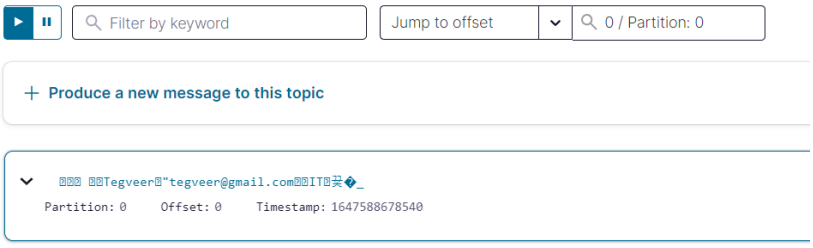
Setting up MySQL source connector



```
(env) C:\Users\tegve\Projects2022\Cloud-Computing\Milestone-3\SOFE4630U-tut3\connectors\mysql>py cons_mysql.py
partition:0
key:
value:
{'id': 1, 'name': 'alice', 'email': 'alice@abc.com', 'department': 'eng.', 'modified': datetime.datetime(2022, 3,

--------

partition:0
key:
value:
{'id': 2, 'name': 'bob1', 'email': 'bob1@abc.com', 'department': 'sales', 'modified': datetime.datetime(2022, 3, 1

--------

partition:0
key:
value:
{'id': 3, 'name': 'bob2', 'email': 'bob2@abc.com', 'department': 'sales', 'modified': datetime.datetime(2022, 3, 1

--------

partition:0
key:
value:
{'id': 4, 'name': 'bob3', 'email': 'bob3@abc.com', 'department': 'sales', 'modified': datetime.datetime(2022, 3, 1

--------

partition:0
key:
value:
{'id': 5, 'name': 'bob4', 'email': 'bob4@abc.com', 'department': 'sales', 'modified': datetime.datetime(2022, 3, 1

--------

partition:0
key:
value:
{'id': 6, 'name': 'bob5', 'email': 'bob5@abc.com', 'department': 'sales', 'modified': datetime.datetime(2022, 3, 1
```

Sink connector to MySQL

Screenshot of the newly added values as messages or as values in the GKE MySQL table




```
tegveer2211@cloudshell:~/SOFE4630U-tut3 (cloud-milestone-3)$ mysql -uuser -pSOFE4630U -h35.226.72.125 <<< "use myDB; show tables;"
mysql: [Warning] Using a password on the command line interface can be insecure.
Tables_in_myDB
ToMySQL
test
tegveer2211@cloudshell:~/SOFE4630U-tut3 (cloud-milestone-3)$ mysql -uuser -pSOFE4630U -h35.226.72.125 <<< "select * from myDB.ToMySQL"
mysql: [Warning] Using a password on the command line interface can be insecure.
id      name    email   department      modified
15      user    test@gmail.com  IT      2022-03-18 07:30:29.083
16      Tegveer tegveer@gmail.com       IT      2022-03-18 07:31:18.362
16      Hello   world@gmail.com IT      2022-03-18 07:31:33.488
16      Hello   world@gmail.com IT      2022-03-18 07:34:35.516
17      Hello   world@gmail.com IT      2022-03-18 07:34:42.316
tegveer2211@cloudshell:~/SOFE4630U-tut3 (cloud-milestone-3)$
```