Kafka-MySQL Connection

Tools used

1) Kafka: 2.13-2.8.0
2) MySQL: 8.0.30

Connectors used

1. mysql-connector-java-8.0.29.jar (platform independent version)
2. confluentinc-kafka-connect-jdbc-10.5.1

Prerequisite

MySQL version 8.0.29 or version 8.0.30 is running on your local machine.

Configurations

Find out your public ip using command in the linux terminal: ifconfig

Inside MySQL

1. Find you MySQL configuration file. Usually it is in /etc/mysql/mysql.conf.d/mysqld.cnf
2. Update MySQL bind-address to 0.0.0.0
(If there is no line like “bind-address = 127.0.0.1”, add “bind-address = 0.0.0.0”)
3. Restart MySQL with updated configuration: sudo systemctl restart mysql.service

Inside Raptor

1. In your input.graphml, “mysqlCfg” attribute is added for providing MySQL configuration at a node. On your desired node, add path to your MySQL file, here “kafka/config/mysql-bulk-sink.properties” is a relative path in the amnis-data-sync directory. You can use own file-path.

```
<key id="mysqlCfg" for="node" attr.name="mysqlConfig" attr.type="string"/>

<node id="h1">
        <data key="mysqlCfg">kafka/config/mysql-bulk-sink.properties</data>
</node>
```

2. your MySQL config file contains following information:

```
name=mysql-bulk-sink
connector.class=io.confluent.connect.jdbc.JdbcSinkConnector

tasks.max=1

connection.url=jdbc:mysql://<your_public_ip>/<database name>

connection.user=<mysql_username>

connection.password=<mysql_password>
```

auto.create=true

topics.regex=<topic_name>

(* change configurations in the <> according to your MySQL connection)

3. Produce data (with schemas specified):
Example data:
{"schema":{"type":"struct","optional":false,"version":1,"fields":
[{"field":"ID","type":"string","optional":true},{"field":"Artist","type":"string","optional":true},
{"field":"Song","type":"string","optional":true}]},"payload":{"ID":"1","Artist":"Rick
Astley","Song":"Never Gonna Give You Up"}}

Here, field and type will specify the column names and data types respectively in your MySQL table. Payload will contain the data for the table rows.

Sample Command

1.Provide MySQL configuration in the node attribute in your input.graphml and run command:
sudo python3 main.py use-cases/testing-applications/input.graphml --nbroker 2 --nzk 1
2. In your Spark application, write your output stream to the topic named <topic_name>
3. As soon as data is ingested to that Kafka topic, your MySQL gets updated in real-time.
4. Once simulation is complete, you will a MySQL table named on <topic_name> in the specified database.