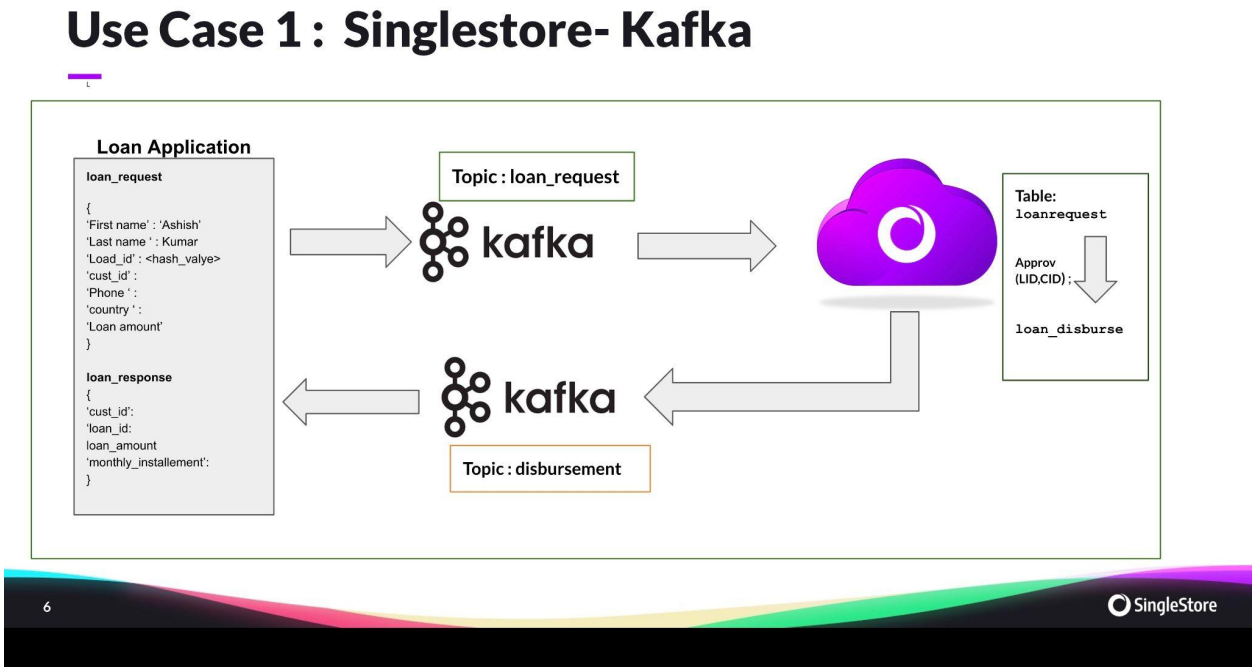
Bi-directional data movement from Kafka to Singlestore.

In this demo we will demonstrate a Loan Processing application Where the user fills in their data and which is sent to Kafka Topic in the form of JSON. The Json data is then pulled by the Signlestore Pipeline and stored in a Table.

With the help of a user defined Procedure the Loan is processed which results in sending the processed information back to the Kafka in the JSON form.The application can consume this JSON as a response to the loan submission after it is processed .



Main components of this application:

1. Producer.py
2. Consumer.py
3. Createtopic.sh
4. Database DDL
5. Disbursement.sh

We are going to explain and build these as we progress.The prerequisite to this demo is to have Python and Kafka installed.

Lets get started :

1.Producer.py

This is a simple python script which simulates the application where users are continuously feeding in data. This script keeps on generating JSON data with a delay of 3-9 sec to give us a real time scenario.

```
import time
import json
import random
from datetime import datetime
from data_generator import generate_message
from kafka import KafkaProducer

# Messages will be serialized as JSON
def serializer(message):
    return json.dumps(message).encode('utf8')

# Kafka Producer
producer = KafkaProducer(
    bootstrap_servers=['localhost:9092'],
    value_serializer=serializer
)
```

```
if __name__ == '__main__':
    # Infinite loop - runs until you kill the program
    while True:
        # Generate a message
        dummy_message = generate_message()

        # Send it to our 'messages' topic
        print(f'Producing message @ {datetime.now()} | Message = {str(dummy_message)}')
        producer.send('loan_request', dummy_message)

        # Sleep for a random number of seconds
        time_to_sleep = random.randint(1, 11)
        time.sleep(time_to_sleep)
```

2.Consumer.py

This script gives the output of the data inserted in the topic this is to validate if the data is generated frequently and pushed to kafka topic

```
import json
from kafka import KafkaConsumer

if __name__ == '__main__':
    # Kafka Consumer
    consumer = KafkaConsumer(
        'loan_request',
        bootstrap_servers='localhost:9092',
        auto_offset_reset='earliest'
    )
    for message in consumer:
        print(json.loads(message.value))
```

3. Createtopic.sh

There are two topics in this use case: loan_request & disbursement. The loan_request is created automatically when we run the data generator app Producer.py , we need to create another topic disbursement which is going to hold the JSON data when we process the Loan.

```
sh /u01/kaf/kafka-3.1.0-src/bin/kafka-topics.sh --create --topic disbursement
--bootstrap-server localhost:9092
```

4. Database DDL

In this section we are going to create tables to store data and write the procedure which will process the Loan.

4.1 Create two tables loanrequest for storing the incoming data & disbursement to store processed data

```
create table loanrequest (cust_ids int,loan_amount int,loan_id TEXT,first_name
VARCHAR(20),last_name VARCHAR(20),country VARCHAR(50),phone TEXT);


create table loan_disburse(cust_ids int,loan_id TEXT,loan_amount int , monthly_inst
int, PRIMARY KEY(cust_ids,loan_id));
```

4.2 In this step we will create a procedure to read the data from pipeline and format that to store in loan request table

```
DELIMITER //
CREATE OR REPLACE PROCEDURE proc_loanreq(GENERIC_BATCH query(GENERIC_JSON
json)) AS
BEGIN
INSERT INTO loanrequest(cust_ids,loan_amount,loan_id,first_name,last_name,country,phone)
SELECT
GENERIC_JSON::cust_ids,GENERIC_JSON::loan_amount,GENERIC_JSON::$loan_id,GENERI
C_JSON::$first_name,GENERIC_JSON::$last_name,GENERIC_JSON::$country,GENERIC_JSO
N::$phone
FROM GENERIC_BATCH;
END //
DELIMITER ;



CREATE PIPELINE kaf_loanreq
AS LOAD DATA KAFKA '44.204.84.174:9092/loan_request'
INTO procedure proc_loanreq (GENERIC_JSON <- %) format json;

START PIPELINE kaf_loanreq;
```

4.3 In this step we will create a procedure which will take cust id and loan id to process the loan . As a processing a row (cust_ids,loan_id,loan_amount,monthly_inst) will be inserted to the disbursement table for the loan id which is processed.The JSON data will be sent to disbursement kafka topic once the row is inserted into disbursement table .There will a calculation of monthly emi in this procedure.

```
DELIMITER //
CREATE OR REPLACE PROCEDURE approve_loan(cid int,lid TEXT) AS
BEGIN
INSERT INTO loan_disburse(cust_ids,loan_id,loan_amount,monthly_inst)
SELECT cust_ids,loan_id,loan_amount,((loan_amount*1.17)/12) from loanrequest where
cust_ids=cid AND lid=loan_id  ;
SELECT TO_JSON(loan_disburse.*) FROM loan_disburse where cust_ids=145 AND
loan_id='GMBPXGhjscTkxrrbVsqNcaHrEEtSeApg'
INTO KAFKA '44.204.84.174/disbursement';
END //
DELIMITER ;
```

4.4 calling the procedure to approve a loan

```
CALL approve_loan(145,'GMBPXGhjscTkxrrbVsqNcaHrEEtSeApg');
```

5. Disbursement.sh

This script will give the final output in JSON format from the Disbursement kafka topic once the loan is approved.

```
sh /u01/kaf/kafka-3.1.0-src/bin/kafka-console-consumer.sh --topic disbursement
--from-beginning --bootstrap-server localhost:9092
```

Testing the completed setup :

1. Start by executing consumer.py in a shell it will show the JSON once the producer.py is started
2. Start producer.py in another shell , you will see the JSON data on consumer.py shell
3. Validate the loanrequest table in the database by executing

*Select * from loanrequest;*

Once you see the table getting frequently updated by the singlestore pipeline pulling data from kafka topic show kafka to singlestore data movement.

4. Start disbursement.sh in a new shell .
5. Pick a loan id & customer id from output of step 3 sql and execute the below

CALL approve_loan(<CUST_ID>,<'LOAN ID'>);

This should give you a row in the disbursement table and JSON data in the disbursement shell opened in step 4.

Validate this by executing the below sql in the database.

*Select * from loan_disburse*

This conclude the bi-drectional movement of data back from singlestore to kafka Topic

Thankyou