

Kafka Zensar Project

Introduction

This project is about building the backend applications that use the Spring Data JPA, Docker, Kafka, and MySQL. In this context, our applications will be running on Docker.

About

This project is about reading a file from the filesystem (directory <json file>). The JSON file is read from the directory, and the content is produced to a Kafka topic. The content is pushed through a REST API. Read the data from Kafka and persist the data into MySQL.

The project consists of the following:

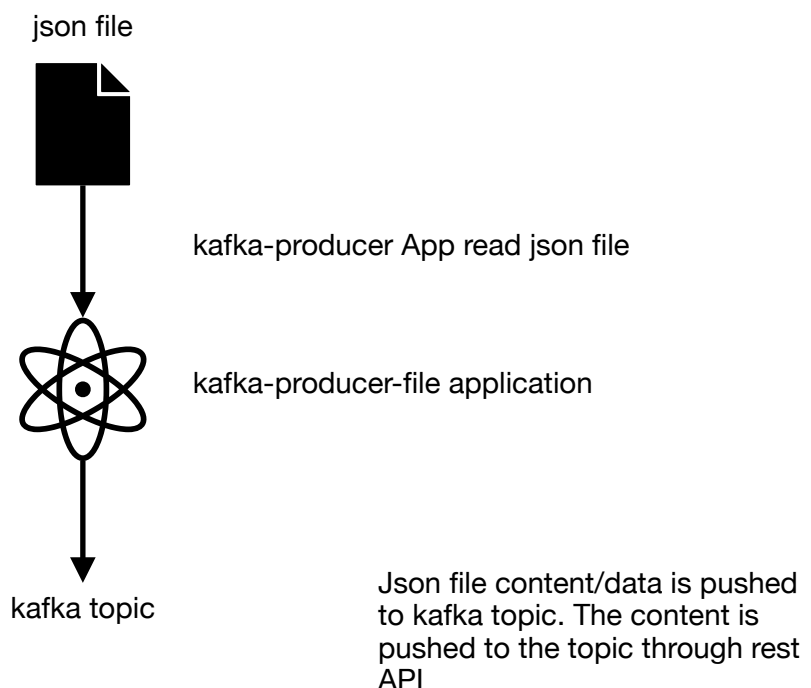
1. <data> directory (with Person.json file)

NB. You can modify the JSON and send

```
{
  "id": "10",
  "gender": "male",
  "firstname": "John",
  "lastname": "Doe"
}
```

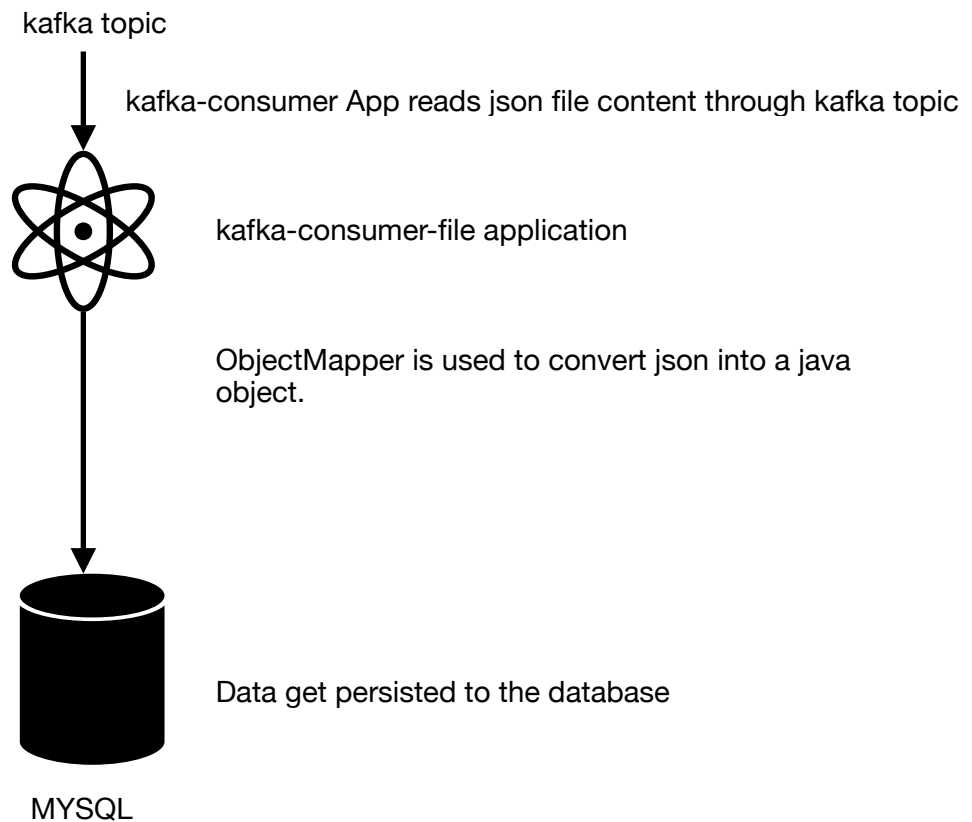
2. kafka-producer-file (Spring Boot application project)

This application reads a JSON file from the above directory and pushes (produces) the JSON data to a Kafka topic.



3. kafka-consumer-file(spring-boot application project)

This application consume json content/data from the kafka-producer-file application through kafka topic and persist the json data to MySQL.



Technologies

- Docker
- Spring Boot
- Apache kafka and zookeeper
- databases(mysql and sqlserver)

Install

- Install JDK java 11
- Install docker
- Eclipse/IDEA/other tools

Running the project

1. Git clone the project

git clone <https://github.com/NhlamuloChauke/kafka-zensar-project.git>

2. Run spring-boot and docker containers(for mysql, kafka, kafka-producer-file-application and kafka-consumer-file-application)

```
cd kafka-zensar-project
ls
```

```
README.md  data  kafka-consumer-file  mysql-init  Zensar Solution
Document.pages  docker-compose.yml  kafka-producer-file
```

```
cd data
cat Person.json
```

```
{
  "id": "1",
  "gender": "male",
  "firstname": "John",
  "lastname": "Doe"
}
```

```
cd ../
```

```
cd kafka-consumer-file
mvn clean install
```

```
cd ../kafka-zensar-project
cd kafka-producer-file
mvn clean install
```

```
cd ../kafka-zensar-project
docker network create docker_default_net ##creating docker network
docker-compose up -d kafka-mysql-db    ## creating MYSQL docker
docker-compose up -d kafka              ## creating kafka docker
docker-compose up -d zookeeper          ## creating zookeeper docker
```

```
### create kafka topics
docker-compose exec kafka bash
bash-5.1#
```

Change the directory to /opt/kafka/bin where you find scripts such as kafka-topics.sh.

```
cd /opt/kafka/bin
```

```
bash-5.1# cd /opt/kafka/bin
bash-5.1# kafka-topics.sh --create --topic person-json-file-topic --
bootstrap-server localhost:9092
```

```
docker-compose build kafka-consumer-file  ## building docker image
for our App
docker-compose up -d kafka-consumer-file  ## running docker for
our App
docker-compose build kafka-producer-file  ## building docker image
for our App
docker-compose up -d kafka-producer-file  ## running docker image
for our App
```

```
docker ps ## list docker containers
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS		
NAMES			
c39309194d2f	kafka-producer-file	"java -jar /app.jar"	4 minutes ago
Up 4 minutes	0.0.0.0:9002->9002/tcp		
kafka-producer-file			
aa9169ed54cb	kafka-consumer-file	"java -jar /app.jar"	25 minutes ago
Up 25 minutes	0.0.0.0:9001->9001/tcp		
kafka-consumer-file			
056bb9b35d1d	mysql/mysql-server:5.7	"/entrypoint.sh mysql..."	45 minutes ago
Up 45 minutes (healthy)	0.0.0.0:3306->3306/tcp, 33060/tcp		
kafka-mysql-db			
5cf3147bc284	wurstmeister/kafka	"start-kafka.sh"	49 minutes ago
Up 49 minutes	0.0.0.0:9092->9092/tcp		
kafka-zensar-project_kafka_1			
ae0a711e7915	wurstmeister/zookeeper	"/bin/sh -c '/usr/sb..."	54 minutes ago
Up 54 minutes	22/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp		
kafka-zensar-project_zookeeper_1			

3. REST API

```
# send file to a kafka topic (kafka-producer-file)
```

```
curl -X POST http://localhost:9002/send
```

```
# get all the persisted data from the database(kafka-consumer-file)
```

```
curl -X GET http://localhost:9001/persons
```

```
[{"id":1,"gender":"male","firstname":"John","lastname":"Doe"},
{"id":2,"gender":"male","firstname":"John","lastname":"Doe"}]
```

