Integration of Apache kafka with Throntail

Durgesh Anaokar (SME Red Hat)
Varsha Kamble (STSE Red Hat)
Agenda

- What is Kafka?
- Terminologies of Kafka.
- Brief about Microprofile.
- Integration of Kafka with Thorntail.
What Is Kafka?

**Kafka Use Cases**

- Messaging
- Stream Processing
- Event Sourcing
- Website Activity tracking
- Log Aggregation
- Commit log
Kafka’s four core APIs

- Producer API
- Consumer API
- Streams API
- Connector API
Apache Kafka Terminology

- Producer
- Consumer
- Broker
- Topic
- Partitions
- Offset
- Consumer groups
Based off Messaging systems

- Producers
- Consumers
- Topics
Kafka record

- Each Record consists of a Key / Value / Timestamp
- Immutable
- Persisted (up to a retention limit)
Kafka Topic

- A topic is a category or feed name to which records are published.
- Topics in Kafka are always multi-subscriber; that is, a topic can have zero, one, or many consumers that subscribe to the data written to it.
Kafka Consumers

- Reads Records from Topics
- Can start at any offset
- Consumer Groups are a logical grouping of Consumers
- Ensures a Record is only read by a single Consumer within a Consumer Group
Kafka Is Polyglot

- Kafka communication from clients and servers uses a wire protocol over TCP that is versioned and documented.
- There are clients in C#, Java, C, Python, Ruby, and many more languages.
- The Kafka ecosystem also provides REST proxy allows easy integration via HTTP and JSON, which makes integration even easier.
MicroProfile
What is Thorntail

- Previously known as WildFly Swarm.
- It's MicroProfile compatible.
- Framework based on the popular WildFly Java application server.
- Standalone microservice-based applications.
- Packaging and running Java EE application with just enough of the server runtime to "java -jar".
CDI Extension for Kafka

- Utilizes Kafka APIs underneath
- Removes boilerplate code for configuring Kafka
- Converts “pull” of Kafka API into “push” for CDI

```xml
<dependency>
  <groupId>org.aerogear.kafka</groupId>
  <artifactId>kafka-cdi-extension</artifactId>
  <version>0.1.0</version>
</dependency>
```
Vanilla Kafka consumer vs CDI Extension

**Vanilla Kafka consumer**

```java
final Properties props = new Properties();
props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "my-cluster-kafka:9092");
props.put(ConsumerConfig.GROUP_ID_CONFIG, "demo-group");
props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
final KafkaConsumer<String, String> consumer = new KafkaConsumer(props);
...
consumer.subscribe(Collections.singleton("topic"));
...
final ConsumerRecords<String, String> records = consumer.poll(500);
for (final ConsumerRecord<String, String> record : records) {
    logger.info(record.value());
}
```

**CDI Extension**

```java
@KafkaConfig(bootstrapServers = ":<KAFKA_SERVICE_HOST>:<KAFKA_SERVICE_PORT>")
public class MyConsumer {
    private static final Logger LOGGER = LoggerFactory.getLogger(MyConsumer.class);
    
    @Consumer(topics = "topic", groupId = "demo-group")
    public void onMessage(final String key, final String value) {
        LOGGER.info("We got this value: " + value);
    }
}
```
Demo
Questions?
References

Kafka: https://kafka.apache.org

Thorntail: https://thorntail.io/

Thank you