Lab 5.2: Adding a clean shutdown to our producer

Welcome to the session 5 lab 2. The work for this lab is done in `~/kafka-training/lab5.2`. In this lab, you are going to create a clean shutdown for our advanced Producer.

Please refer to the Kafka course notes for any updates or changes to this lab.

Find the latest version of this lab here.

Lab Adding an orderly shutdown flush and close

Shutdown Producer Nicely

Let's write some code to shut the Producer down nicely.

The shutdown code will happen if you are running the producer example from a terminal and type ctrl-C so shutdown from Java occurs. We will write some code to shutdown thread pool and wait. Then we will flush the producer to send any outstanding batches if using batches (producer.flush()). Lastly, we will close the producer using producer.close and wait five seconds for the producer to shutdown. (Note that closing the producer also flushes it.)

To this we will add shutdown hook to Java runtime. Then to test we will start the StockPriceKafkaProducer, and then you can stop it using CTRL-C or by pressing the stop button in your IDE.

~/kafka-training/lab5.2/src/main/java/com/cloudurable/kafka/producer/StockPriceKafkaProducer.java

Kafka Producer: StockPriceKafkaProducer shutdown hook for clean shutdown

```
public class StockPriceKafkaProducer {

    ...

    private static final Logger logger =
LoggerFactory.getLogger(StockPriceKafkaProducer.class);



    public static void main(String... args) throws Exception {
        //Create Kafka Producer
        final Producer<String, StockPrice> producer = createProducer();
        //Create StockSender list
        final List<StockSender> stockSenders = getStockSenderList(producer);

        //Create a thread pool so every stock sender gets it own.
        // Increase by 1 to fit metrics.
        final ExecutorService executorService =
                Executors.newFixedThreadPool(stockSenders.size() );

        //Run each stock sender in its own thread.
        stockSenders.forEach(executorService::submit);


        //Register nice shutdown of thread pool, then flush and close producer.
        Runtime.getRuntime().addShutdownHook(new Thread(() -> {
            executorService.shutdown();
            try {
                executorService.awaitTermination(200, TimeUnit.MILLISECONDS);
                logger.info("Flushing and closing producer");
                producer.flush();
```

```
                producer.close(10_000, TimeUnit.MILLISECONDS);
            } catch (InterruptedException e) {
                logger.warn("shutting down", e);
            }
        }));
    }

...
}
```

Notice we add a shutdown hook using Runtime.getRuntime().addShutdownHook and this shutdown hook that shuts down the thread pool, then calls flush on the producer and then closes the producer whilst waiting 10 seconds for the close to happen.

ACTION - EDIT src/main/java/com/cloudurable/kafka/producer/StockPriceKafkaProducer.java and add a shutdown hook to main.

ACTION - RUN this StockPriceKafkaProducer and try shutting it down.

Kafka Tutorial

This comprehensive *Kafka tutorial* covers Kafka architecture and design. The *Kafka tutorial* has example Java Kafka producers and Kafka consumers. The *Kafka tutorial* also covers Avro and Schema Registry.

Complete Kafka Tutorial: Architecture, Design, DevOps and Java Examples.

- Kafka Tutorial Part 1: What is Kafka?
- [Kafka Tutorial Part 2: Kafka Architecture](http://cloudurable.com/blog/kafka-architecture/index.html "This Kafka tutorial discusses the structure of Kafka. Kafka consists of Records, Topics, Consumers, Producers, Brokers, Logs, Partitions, and Clusters. Records can have key, value and timestamp. Kafka Records are immutable. A Kafka Topic is a stream of records - "/orders", "/user-signups". You can think of a Topic as a feed name. It covers the structure of and purpose of topics, log, partition, segments, brokers, producers, and consumers")
- Kafka Tutorial Part 3: Kafka Topic Architecture
- Kafka Tutorial Part 4: Kafka Consumer Architecture
- Kafka Tutorial Part 5: Kafka Producer Architecture
- Kafka Tutorial Part 6: Using Kafka from the command line
- Kafka Tutorial Part 7: Kafka Broker Failover and Consumer Failover
- Kafka Tutorial Part 8: Kafka Ecosystem
- Kafka Tutorial Part 9: Kafka Low-Level Design
- Kafka Tutorial Part 10: Kafka Log Compaction Architecture
- Kafka Tutorial Part 11: Writing a Kafka Producer example in Java
- Kafka Tutorial Part 12: Writing a Kafka Consumer example in Java
- Kafka Tutorial Part 13: Writing Advanced Kafka Producer Java examples
- Kafka Tutorial 14: Writing Advanced Kafka Consumer Java examples
- Kafka Tutorial Part 15: Kafka and Avro
- Kafka Tutorial Part 16: Kafka and Schema Registry
- Kafka Tutorial

About Cloudurable

We hope you enjoyed this article. Please provide feedback. Cloudurable provides Kafka training, Kafka consulting, Kafka support and helps setting up Kafka clusters in AWS.