Kafka Course May 2018

Check back at the [course notes page for updates, notes, and answer to questions during the course](#).

Review the course outline: [Course outline](#)

Files

- The *course files* can be found [here](#).
- The Linux VirtualBox OVA image from [here](#).

The *Virtual Box image* and the *course files zip* has *Kafka* and *Confluent (Schema Registry, etc.)* already installed.

Source, Kafka, Labs, Zip File

The zip file should be exracted to `~/` such that it creates a directory called `~/kafka-training`. The VirtualBox image already has the zip file extracted in the right location. You should use the Kafka VirtualBox image for all labs. [Disclaimer about not using Virtual Box image](#).

Lab files

Every lab folder has a `README.md` with instructions on how to complete the lab, and a `solution` folder with a working version of the lab for reference. Please use the README.md and solution folder as a guide to completing the labs.

System Requirements

The development computer will need 16 GB of RAM, and should have 2.5 GHz Intel Core i7 or better for the least amount of frustration. Most developer computers will have this much or more. You should have 30 GB of free disk space. You will need to run bash scripts. These have been tested under Linux and OSX.

Windows Users

We recommend using the Linux Virtual machine image if you are using Windows. We don't support running the labs on Windows direct or Windows batch files. You will need to install VirtualBox 5.1.x. We used VirtualBox 5.1.22 which is the latest as of July 17th, 2017. Use the provided image above, which has Java 1.8, Kafka, and Confluent Streams already installed. We can't stop the course to help install Kafka, Java, Gradle, Scala, etc. on 15+ developer boxes. Please use the image.

OSX Users and Linux Users

You can just unzip the zip file under `~/` so that it creates a directory `~/kafka-training`. You need the Java 1.8 JDK and a Java IDE of your choice. The zip file is self-contained (other than the JDK) so the environment issues should be minimal.

However, if you have environment problems, please use the VirutalBox image for the course instead. We used [VirtualBox 5.X.X](#). Use the provided image above, which has IDEA, Java 1.8, Kafka, and Confluent Streams already installed. *There are thirty labs and 15 students (and four days) so we will not have a lot of time to sort out environment issues on 15 development boxes.* The focus will be on Kafka not your environment.

The VirtualBox image uses Ubuntu and it was installed via [these steps](#). You do not have to install. Just use the image.

[Go here to install VirtualBox](#). Once virtual box is installed, you can double click the OVA image file from above and the image will be installed. You may want to increase the RAM to 8GB and the number of processors used by the

image to 4.

Lab Instructions

The latest instructions for the labs can be found below.

1. [Lab 1](#) Basic Kafka
2. [Lab 1.2](#) Understanding Failover
3. [Lab 2](#) Writing a simple Java Producer
4. [Lab 3](#) Writing a simple Java Consumer
5. [Lab 5.1](#) Advanced Producers
6. [Lab 5.2](#) Clean shutdown
7. [Lab 5.3](#) Configuring Producer Durability
8. [Lab 5.4](#) Metrics and Replication Verification
9. [Lab 5.5](#) Kafka Batching Records (compression)
10. [Lab 5.6](#) Retry, timeouts, in-flight, back-off
11. [Lab 5.7](#) Kafka Interceptor (Optional)
12. [Lab 5.8](#) Kafka Custom Partitioner (Optional)
13. [Lab 6.1](#) Advanced Consumer
14. [Lab 6.2](#) Position Consumer in Topic Log
15. [Lab 6.3](#) At-Most-Once and At-Least-Once Semantics
16. [Lab 6.4](#) Exactly-Once message semantics
17. [Lab 6.5](#) Thread per consumer
18. [Lab 6.6](#) Consumer with worker threads
19. [Lab 6.7](#) Priority queue with consumer.assign
20. [Lab 7.1](#) Avro
21. [Lab 7.2](#) Schema Registry
22. [Lab 8.1](#) Kafka Security SSL
23. [Lab 8.3](#) Kafka Security SASL PLAIN
24. [Lab 8.4](#) Kafka Security SASL SCRAM
25. [Lab 9](#) Kafka MirrorMaker
26. Lab 10 DEMO walk through
27. [Lab Streams 1](#) Kafka Streams Lab 1
28. [Lab Streams 2](#) Kafka Streams Lab 2