Deploying a Kafka Cluster: Demos Guide

This short document describes the steps necessary to run the demos examples.

Pre-requisites

Java

The code examples in this course are provided in Java (version 11). Even if you don't know Java, the examples are simple enough that knowing any other programming language you should quickly get used to the syntax. To be able to run the examples, please install the Java Development Kit (JDK 11). You can do this by following instructions for any of the available Java distributions. I suggest using AdoptOpenJDK (https://adoptopenjdk.net/) or Zulu Community from Azul Systems (https://www.azul.com/downloads/zulu-community/). After the installation, you should be able to run `java -version` without getting any errors.

Docker and Docker Compose

To run Kafka and other parts of the system in an isolated fashion, I will use Docker containers. To install Docker and Docker Compose on Windows or Mac, you should install Docker Desktop (https://docs.docker.com/desktop/). To install it on Linux, install the Docker engine (https://docs.docker.com/engine/install/) and then Docker Compose (https://docs.docker.com/compose/install/). Please verify that you are able to run `docker` and `docker-compose` in the terminal.

Editing code and running the examples

All the examples in the course are ran from the command line. So you can use any Editor/IDE that you like - this will not impact your ability to run the examples. Most of the time I will run one of two commands:

- `docker-compose` – to bring up and manage the containers
- `./gradlew` – Invoke the Gradle build system commands in order to build the Java code and execute it. Note that to run Gradle you only need to have JDK installed. You don't have to really understand Gradle – most of the commands will be in the format `./gradlew :moduleN:run -PmainClass='mN.FileName` where N stands for the module number. This command will execute the 'main' method in the module's N FileName.

Directory structure

The 'Materials.zip' contains two directories: 'before' and 'after'. 'Before' contains the exercise files with some gaps to fill (you will see me doing that throughout the course), and 'after' contains finished exercises. I suggest working with the 'before' version and check against 'after'. Let's assume that the 'before' directory is the demos root. Each module has its own directory under the demos root. Under each module, there is a 'docker' directory that contains everything container-related, and 'src', that contains Java files. When running the `./gradlew` command please make sure to do that from the demos root directory, while when running `docker-compose` please make sure that you are in the module's 'docker' directory.

Should you have any questions, don't hesitate to post them on the course discussion. Enjoy the course!