Day 1 – Lab1:

Hello World, Kafka

In this lab, you will install Kafka with Docker and verify it is working by creating a topic and sending some messages.

Objectives

1. Install Kafka using Docker
2. Create a topic
3. Send some messages to the topic
4. Start a consumer and retrieve the messages

Prerequisites

One of the easiest way to get started with Kafka is through the use of Docker. Docker allows the deployment of applications inside software containers which are self-contained execution environments with their own isolated CPU, memory, and network resources. Install Docker Desktop by following the directions appropriate for your operating system. Make sure that you can run both the `docker` and `docker-compose` command from the terminal.

Instructions

1. Open a terminal in this lab directory: `labs/01-Verify-Installation`.

2. Start the Kafka and Zookeeper processes using Docker Compose:

```
$ docker-compose up
```

The first time you run this command, it will take a while to download the appropriate Docker images.

3. Open an additional terminal window in the lesson directory, `labs/01-Verify-Installation`. We are going to create a topic called `helloworld` with a single partition and one replica:

```
$ docker-compose exec kafka kafka-topics.sh --bootstrap-server :9092 --create --replication-factor 1 --partitions 1 --topic helloworld
```

4. You can now see the topic that was just created with the `--list` flag:

```
$ docker-compose exec kafka kafka-topics.sh --bootstrap-server :9092 --list
helloworld
```

NOTE: If you see `__consumer_offsets`, it is an administrative topic automatically created by Kafka itself.

5. Normally you would use the Kafka API from within your application to produce messages but Kafka comes with a command line *producer* client that can be used for testing purposes. Each line from standard input will be treated as a separate message. Type a few messages and leave the process running.

```
$ docker-compose exec kafka kafka-console-producer.sh --bootstrap-server :9092 --topic
helloworld
Hello world!
Welcome to Kafka.
```

NOTE: use keystroke `ctrl-d` to end message production via the terminal.

6. Open another terminal window in the lesson directory. In this window, we can use Kafka's command line *consumer* that will output the messages to standard out.

```
$ docker-compose exec kafka kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic
helloworld --from-beginning
Hello world!
Welcome to Kafka.
```

7. In the *producer* client terminal, type a few more messages that you should now see echoed in the *consumer* terminal.

8. [OPTIONAL] You may want to try a bit more text to see how Kafka is able to keep up with a larger load of text. For example, you may try to paste the complete work of "War and Peace". You can find the text here: https://www.gutenberg.org/files/2600/2600-0.txt. To do so, simply copy the complete text from a web browser and paste it into the kafka-producer. You may notice that the consumer is processing the text in batches as well as having no problem keeping up with the paste speed of your terminal.

9. Stop the producer and consumer terminals by issuing `ctrl-c`.

10. Finally, stop the Kafka and Zookeeper servers with Docker Compose:

11. `$ docker-compose down`

Congratulations, this lab is complete!