Hands-on Lab: Working with Streaming Data using Kafka

Estimated time needed: **20** minutes

Introduction

In this lab, you will work with streaming data using Kafka. You will start by configuring the Kafka server to use the Kraft mode followed by starting the Kafka message broker service, creating a topic and then starting the producer and consumer.

Objectives

After completing this lab, you will be able to:

- Download Kafka binaries
- Configure the Kafka server to use the KRaft mode
- Start the Kafka message broker service
- Create a topic
- Start a producer
- Start a consumer

About Skills Network Cloud IDE

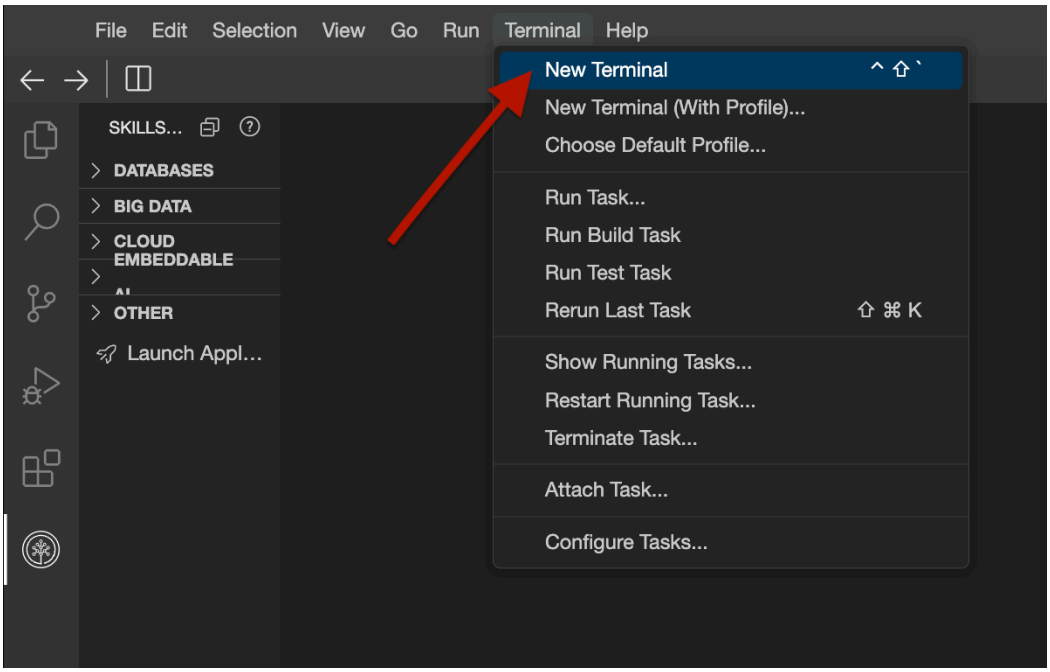
Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands-on labs for course and project-related labs. Theia is an open-source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia running in a Docker container.

Important notice about this lab environment

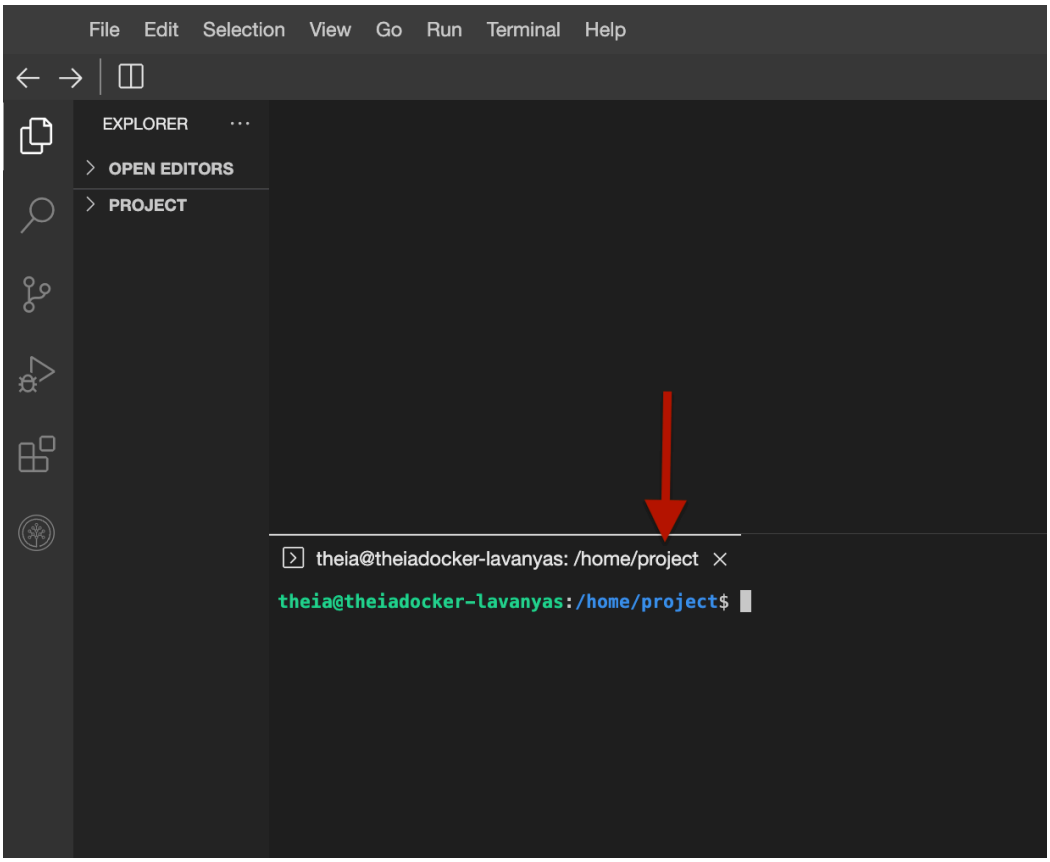
Please be aware that sessions for this lab environment are not persistent. A new environment is created for you every time you connect to this lab. Any data you may have saved in an earlier session will get lost. To avoid losing your data, please plan to complete these labs in a single session.

Exercise 1: Download and extract Kafka

1. Open a new terminal by clicking the menu bar and selecting **Terminal->New Terminal**, as shown in the image below.



This will open a new terminal at the bottom of the screen.



Next, run the following commands on the terminal.

2. Download Kafka by running the command below:

```
wget https://downloads.apache.org/kafka/3.8.0/kafka_2.13-3.8.0.tgz
```

3. Extract Kafka from the zip file by running the command below.

```
tar -xzf kafka_2.13-3.8.0.tgz
```

This command will create a new directory `kafka_2.13-3.8.0` in the current directory.

Exercise 2: Configure KRaft and start server

1. Navigate to the `kafka_2.13-3.8.0` directory.

```
cd kafka_2.13-3.8.0
```

2. Generate a cluster UUID that will uniquely identify the Kafka cluster.

```
KAFKA_CLUSTER_ID="$(bin/kafka-storage.sh random-uuid)"
```

This cluster id will be used by the KRaft controller.

3. KRaft requires the log directories to be configured. Run the following command to configure the log directories passing the cluster ID.

```
bin/kafka-storage.sh format -t $KAFKA_CLUSTER_ID -c config/kraft/server.properties
```

4. Now that KRaft is configured, you can start the Kafka server by running the following command.

```
bin/kafka-server-start.sh config/kraft/server.properties
```

You can be sure that the Kafka server has started when the output displays messages like "Kafka Server started".

```
[2024-06-12 02:19:51,129] INFO [BrokerServer id=1] Transition from STARTING to STARTED (kafka.server.BrokerSer
ver)
[2024-06-12 02:19:51,130] INFO Kafka version: 3.7.0 (org.apache.kafka.common.utils.AppInfoParser)
[2024-06-12 02:19:51,135] INFO Kafka commitId: 2ae524ed625438c5 (org.apache.kafka.common.utils.AppInfoParser)
[2024-06-12 02:19:51,135] INFO Kafka startTimeMs: 1718173191129 (org.apache.kafka.common.utils.AppInfoParser)
[2024-06-12 02:19:51,137] INFO [KafkaRaftServer nodeId=1] Kafka Server started (kafka.server.KafkaRaftServer)
[2024-06-12 02:20:25,678] INFO [ReplicaFetcherManager on broker 1] Removed fetcher for partitions Set(bankbran
ch-1, bankbranch-0) (kafka.server.ReplicaFetcherManager)
[2024-06-12 02:20:25,718] INFO [LogLoader partition=bankbranch-1, dir=/tmp/kraft-combined-logs] Loading produc
er state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
[2024-06-12 02:20:25,722] INFO Created log for partition bankbranch-1 in /tmp/kraft-combined-logs/bankbranch-1
 with properties {} (kafka.log.LogManager)
[2024-06-12 02:20:25,725] INFO [Partition bankbranch-1 broker=1] No checkpointed highwatermark is found for pa
rtition bankbranch-1 (kafka.cluster.Partition)
[2024-06-12 02:20:25,727] INFO [Partition bankbranch-1 broker=1] Log loaded for partition bankbranch-1 with in
itial high watermark 0 (kafka.cluster.Partition)
[2024-06-12 02:20:25,745] INFO [LogLoader partition=bankbranch-0, dir=/tmp/kraft-combined-logs] Loading produc
er state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
[2024-06-12 02:20:25,746] INFO Created log for partition bankbranch-0 in /tmp/kraft-combined-logs/bankbranch-0
 with properties {} (kafka.log.LogManager)
```

Exercise 3: Create a topic and start producer

You need to create a topic before you can start to post messages.

1. Start a new terminal and change to the `kafka_2.13-3.8.0` directory.

```
cd kafka_2.13-3.8.0
```

2. To create a topic named `news`, run the command below.

```
bin/kafka-topics.sh --create --topic news --bootstrap-server localhost:9092
```

You will see the message: `Created topic news.`

3. You need a producer to send messages to Kafka. Run the command below to start a producer.

```
bin/kafka-console-producer.sh   --bootstrap-server localhost:9092   --topic news
```

4. After the producer starts, and you get the '>' prompt, type any text message and press enter. Or you can copy the text below and paste. The below text sends three messages to Kafka.

```
Good morning
Good day
Enjoy the Kafka lab
```

Exercise 4: Start Consumer

You need a consumer to read messages from Kafka.

1. Start a new terminal and change to the `kafka_2.13-3.8.0` directory.

```
cd kafka_2.13-3.8.0
```

2. Run the command below to listen to the messages in the topic `news`.

```
bin/kafka-console-consumer.sh   --bootstrap-server localhost:9092   --topic news   --from-beginning
```

3. You should see all the messages you sent from the producer appear here.
4. You can go back to the producer terminal and type some more messages, one message per line, and you will see them appear here.

Exercise 5: Explore Kafka directories

Kafka uses the **/tmp//tmp/kraft-combined-logs** directory to store the messages.

1. Start a new terminal and navigate to the `kafka_2.13-3.8.0` directory.

```
cd kafka_2.13-3.8.0
```

2. Explore the root directory of the server.

```
ls
```

3. Notice there is a `tmp` directory. The `kraft-combine-logs` inside the `tmp` directory contains all the logs. To check the logs generated for the topic `news` run the following command:

```
ls /tmp/kraft-combined-logs/news-0
```

Note: All messages are stored in the `news-0` directory under the **/tmp/kraft-combined-logs** directory.

Exercise 6: Clean up

To stop the producer

In the terminal where you are running producer, press `CTRL+C`.

To stop the consumer

In the terminal where you are running consumer, press `CTRL+C`.

To stop the Kafka server

In the terminal where you are running Kafka server, press `CTRL+C`.

Practice exercises

1. Create a new topic named `weather`.

▼ Click here for a **hint**.

Use `kafka-topics.sh` command with the `create` option.

▼ Click here for the **solution**.

Make sure that the Kafka server is still running. Change to the `kafka_2.13-3.8.0` directory and run the following command:

```
bin/kafka-topics.sh --create --topic weather --bootstrap-server localhost:9092
```

2. Post messages to the topic `weather`.

▼ Click here for a **hint**.

Use `kafka-console-producer.sh` and point to topic `weather`.

▼ Click here for the **solution**.

Make sure that the Kafka server is still running. Run the following command:

```
bin/kafka-console-producer.sh   --bootstrap-server localhost:9092   --topic weather
```

Post some test messages.

3. Read the messages from the topic `weather`.

▼ Click here for a **hint**.

Use `kafka-console-consumer.sh` and read from the topic 'weather'.

▼ Click here for the **solution**.

Make sure that the Kafka server is still running. In a new terminal, change to `kafka_2.13-3.8.0` directory and run the following command:

```
bin/kafka-console-consumer.sh   --bootstrap-server localhost:9092   --topic weather
```

Make sure that the messages you sent from the producer appear here.

Authors

Lavanya T S

Other Contributors

Rav Ahuja