THINKPORT

CHEATSHEET

kafka

Setup a dev environment

- To setup a dev environment, clone the following repository and run **docker compose up -d** inside the cp-all-in-one-kraft folder. You can use **localhost:9092** as your bootstrap server using the default settings. You can also access the schema registry via **localhost:8081**.
- To get started using the commands, download the latest version of kafka from here, unpack using **tar -xzf kafka*.tgz**, and add the bin folder to your PATH.

Topics

```
kafka-topics --bootstrap-server localhost:9092 --list
```

- List all topics in your kafka cluster

```
kafka-topics --bootstrap-server localhost:9092 --list
--exclude-internal
```

- List all topics, but exclude all internal topics like __consumer_offsets

Topics

```
kafka-topics --bootstrap-server localhost:9092 --create
--topic YOUR_TOPIC --partitions 3 --replication-factor 1
```

▶Create a topic with three partitions and a replication factor of one.

```
kafka-topics --bootstrap-server localhost:9092 --delete
--topic YOUR_TOPIC
```

▶Delete a topic

```
kafka-topics --bootstrap-server localhost:9092 --describe
--topic YOUR_TOPIC
```

▶Get detailed information on a topic and the partitions (e.g: leader of each partition and number of in-sync replicas)

```
kafka-topics --bootstrap-server localhost:9092 --alter
--topic YOUR_TOPIC --partitions 5
```

▶Alter the number of partitions. Can only be increased. Caution: Messages with keys might be produced to different topics. This could change your ordering.

Consuming & Producing

```
kafka-topics --bootstrap-server localhost:9092 --describe
--topic YOUR_TOPIC
```

▶Consume messages from a topic.

```
kafka-console-consumer --bootstrap-server localhost:9092
--topic YOUR_TOPIC --from-beginning
```

▶Consume messages from a topic and also print all messages from beginning.

Consuming & Producing

```
kafka-console-producer --broker-list localhost:9092
--topic YOUR_TOPIC
```

▶Opens a prompt which takes values of new messages and produces them to the topic.

```
kafka-console-producer --broker-list localhost:9092
--topic YOUR_TOPIC --property "parse.key=true" --property
"key.separator=:"
```

▶Opens a prompt which takes a string in the format "KEY:VALUE" and produces the message with the parsed key and value to the topic.

```
kafka-avro-console-consumer \
--bootstrap-server localhost:9092 \
--topic YOUR_TOPIC \
--property schema.registry.url=http://localhost:8081 \
--property print.key=true --property key.separator=":"
```

▶Consumes messages in the AVRO format. Also prints the key.

```
kafka-avro-console-producer \
--bootstrap-server localhost:9092 \
--topic mytest \
--property schema.registry.url=http://localhost:8081
--property value.schema.id=1
```

▶Opens a prompt which let's you produce AVRO messages to a topic. Message has to be in JSON format. The message will be checked against the specified schema.