**Kafka Practice: Notes and Guidelines**

## CONTENTS

[Practice environment 2](#_Toc5960)

[Kafka topic management 2](#_Toc5961)

[Reading and writing with Kafka topics 2](#_Toc5962)

[Notes regarding Zookeeper 2](#_Toc5963)

[Notes regarding Kafka Connect 3](#_Toc5964)

[Kafka Connect REST Interface 4](#_Toc5965)

[Troubleshooting 4](#_Toc5966)

PRACTICE ENVIRONMENT

Before starting the practice be sure to follow all instructions in the installation and configuration guide of the practice environment. Unsure that all the required services are up and running.

KAFKA TOPIC MANAGEMENT

All scripts for Kafka usage are located in the Kafka bin directory.

* + In your environment it is **/opt/kafka\_2.12-2.4.0/bin/**, which you have added to your PATH environment variable, so you should be able to run the Kafka scripts easily without referring to their location.
  + The default Zookeeper port is 2181.

You can use the **kafka-topics.sh** script to work with Kafka topics.

* + The first parameter to the script should reference your Zookeeper, which runs on **localhost** and listens on the default port.
  + For example, the -**-list** flag, will list all existing topics.

READING AND WRITING WITH KAFKA TOPICS

You can use the **kafka-console-consumer.sh** script to read data from a Kafka topic.

* The Kafka Broker default port is 9092

Note:

* You will only get new messages on the consumer side.

If you wish to get all messages stored in Kafka from the beginning of the queue, specify the flag **--frombeginning** when creating the consumer.

NOTES REGARDING ZOOKEEPER

* The default Zookeeper port is 2181.
* When you refer to Zookeeper in scripts, such as **kafka-topics.sh**, you have two options to reference Kafka topics:
  + In case Zookeeper handles several services (not only Kafka):

**<hostname>:2181/kafka**

* + In case Zookeeper only handles Kafka:

**<hostname>:2181**

NOTES REGARDING KAFKA CONNECT

* All required scripts are located in **$KAFKA\_HOME/bin** and are already part of the **$PATH** in your environment.
* The most suitable script for our practice environment is **connect-standalone.sh**. It accepts few important parameters:
* The Kafka Connect Standalone general properties files
* The file source properties file (where we read from)
* The file sink properties file (in case we are writing to a file destination) **connect-standalone.sh \**

**<connect-standalone-properties-file> \**

**<source-properties>**

**<sink-properties>**

connect-standalone.sh config/connect-standalone.properties config/connect-file-source.properties config/connect-file-sink.properties

Hello! Can you try *file=kafka/srcFiles/newSourceFile.log* instead of ~~file=/kafka/srcFiles/newSourceFile.log~~, also for *file=/kafka/sinkFiles/targetFile.log*? When I tried to run 3rd task 2nd time I faced this problem. Maybe it will help

Hello! Can you see messages like that:

[2024-01-10 09:45:08,951] INFO WorkerSourceTask{id=kafka-file-source-task-2 -0} flushing 0 outstanding messages for offset commit (org.apache.kafka.connect.runtime.WorkerSourceTask:433)  
[2024-01-10 09:45:18,951] INFO WorkerSourceTask{id=kafka-file-source-task-2 -0} Committing offsets (org.apache.kafka.connect.runtime.WorkerSourceTask:416)  
[2024-01-10 09:45:18,952] INFO WorkerSourceTask{id=kafka-file-source-task-2 -0} flushing 0 outstanding messages for offset commit (org.apache.kafka.connect.runtime.WorkerSourceTask:433)

* Files names and locations:
* The Kafka Connect properties file is located in the Kafka configurations directory:

**$KAFKA\_HOME/config/connect-standalone.properties**

* The source and destination properties files, which we create manually as part of the practice are located in the following directory:

**/kafka/confFiles/**

For example:

**/kafka/confFiles/connect-file-source.properties /kafka/confFiles/connect-file-sink.properties /kafka/confFiles/connect-jdbc-source.properties etc..**

* Common connector classes:
* To read and write from files you need to use the **FileStream** connector class:
* **FileStreamSource** for reading from a source file
  + - **FileStreamSink** for writing to a target file
* To read and write from a relational database (such as MySQL) you can use the JDBC connector class:
  + - **io.confluent.connect.jdbc.Jdbc*Source*Connector** for reading
    - **io.confluent.connect.jdbc.Jdbc*Sink*Connector** for writing
* JDBC notes for the practice environment
  + - In the practice environment, be sure to use the network gateway IP address when connecting to a database in a remote Docker container.
    - It is also required to set the parameter **allowPublicKeyRetrieval=true** as a part of the connection string.

This is the part of the initial script you have in the practice document.

KAFKA CONNECT REST INTERFACE

Use an HTTP tool such as a “curl” to work with the REST API of Kafka Connect.

Note:

* It uses port 8083
* Access is via **<host-name>:8083**
* Add it on the machine if it is missing using **apk add curl**

See the documentation for the REST API here: <https://docs.confluent.io/current/connect/references/restapi.html>

It is advised to pipe the result to the Python’s **json.tool** to get more readable results.

For example:

* Listing active connectors on a worker:

**curl localhost:8083/connectors | python -m json.tool**

* Pick a connector from the list and get any information about it, for example:

..**/connectors/<connector>** --connector configuration

..**/connectors/<connector>/tasks** –active tasks

..**/connectors/<connector>/status** –current status

* You can use **kafka-consumer-groups.sh** to reset a topic offset for a particular consumer group. For example, reset to the earliest point possible: **kafka-consumer-groups.sh \**

**--bootstrap-server localhost:9092 \**

**--group** <group-name> **\**

**--reset-offsets --to-earliest \**

**--topic** <topic-name> **\**

**--execute**

TROUBLESHOOTING

Issue:

Error when running the script **bash: *<script-name>*: command not found**

Resolution:

Make sure you add the **.sh** extension to the script when running it. For example, **kafka-topics.sh** instead of just **kafka-topics**

Issue:

* Error **Replication factor: 1 larger than available brokers: 0** while creating a new Kafka topic or
* Error **Topic '<topic-name>' does not exist as expected** when working with a Kafka topic

Resolution:

* See the Zookeeper notes above.
* As Zookeeper handles only Kafka in this Docker, use the Zookeeper root (localhost:2181) and do not reference Kafka specifically (localhost:2181/kafka).

Issue:

Kafka topic is not dropped, even minutes after dropping it

Resolution:

Make sure you exited all consumer and producer processes, which keeps the topic locked.

Issue:

Error **Failed to bind .. 8083 Address in use**

Resolution:

* You probably have an active Kafka connect running.
* Find it and exit it.
* If you cannot locate it, stop and restart the Kafka container: **docker stop kafka docker start kafka**