Capgemini

Apache Kafka Lab Exercises

Document Revision History

Date	Revision No.	Author	Summary of Changes
04-AUG-2021	1.0	Bhimappa Desai	Prepared Lab Exercise Book

Capgemini

Contents

Apache Kafka Installation Steps 3

Setting up Log Directory Steps 4

Creating Kafka-Topics Steps 5

Producing Messages using console-producer - Steps 6

Consuming Messages using console-consumer (from other terminal) - Steps 8

Shifting Topic Off-set - Steps 10

Working with Multi-Broker Cluster - Steps 12

Working with Topic Configuration - Steps 13

Working with Apache Kafka Streaming (Windows Only) - Steps 15

Capgemini

Apache Kafka Installation Steps

##################################

DOWNLOAD KAFKA BINARIES

##################################

1. Download Kafka at **https://kafka.apache.org/downloads**

2. Extract Kafka using 7Zip or WinRAR. Install from following path if not already installed on your system

https://www.7-zip.org/

3. Place Apache Kafka at **C:\kafka_2.12-2.0.0**

4. Install Java 8 JDK:

#http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

5. Open a command line (cmd) type following command to check java version.

this should show JDK 8

C:> **java -version**

6. Try out a following Kafka command from C:\kafka_2.12-2.0.0\bin\windows directory

C:\kafka_2.12-2.0.0\bin\windows**kafka-topics.bat**

7. Edit your environment variables and add below path to PATH variable

C:\kafka_2.12-2.0.0\bin\windows

8. Open a new terminal and Try running the following command from any directory:

kafka-topics.bat

Setting up Log Directory Steps

1. Create Zookeeper and Kafka Directory

C:/kafka_2.12-2.0.0/data/zookeeper

C:/kafka_2.12-2.0.0/data/kafka

2. Edit zookeeper.properties file from configuration directory using notepad Using NotePad++ if not installed already, download and install from this link https://notepad-plus-plus.org/download/v7.5.8.html

change **dataDir** Properties in zookeeper.properties file

 # dataDir=C:/kafka_2.12-2.0.0/data/zookeeper

3. Start zookeeper (make sure nothing is running on port 2181)

Change Directory to C:\kafka_2.12-1.0.0\bin\windows

4. Enter following command##

C:\kafka_2.12-1.0.0\bin\windows>zookeeper-server-start.bat \kafka_2.12-1.0.0\config\zookeeper.properties

5. Open a new command line (we leave zookeeper running in previous command line)

Edit server.properties file from configuration directory and change **log.dirs** properties in server.properties file

 # log.dirs=C:/kafka_2.12-2.0.0/data/kafka

6. Change Directory to **C:\Kafka_2.12-1.0.0\bin\windows** and Start Kafka

7. Enter following command

C:\kafka_2.12-1.0.0\bin\windows>**kafka-server-start.bat \kafka_2.12-1.0.0\config\server.properties**

7. Kafka is running! And Keep the two command line windows opened

Creating Kafka-Topics Steps

1. By "kafka-topics.sh" or "kafka-topics.bat" based on your system # (or bin/kafka-topics.sh or bin\windows\kafka-topics.bat if you didn't setup PATH / Environment variables. If you have set up a path variable, then just type kafka-topics from any directory)

2. Just to see help about kafka-topics enter following command
   kafka-topics

3. Enter following command to list of existing topics

   kafka-topics --zookeeper 127.0.0.1:2181 --list

4. Enter following command to create new topics (first-topic) and evaluate the message.

   kafka-topics --zookeeper 127.0.0.1:2181 --topic first-topic --create

5. Enter following command to create new topics (first-topic) and evaluate the message.

   kafka-topics --zookeeper 127.0.0.1:2181 --topic first-topic --create --partitions 3

6. Enter following command to create new topics (first-topic) and evaluate the message.

   kafka-topics --zookeeper 127.0.0.1:2181 --topic first-topic --create --partitions 3 --replication-factor 2

7. Enter following command to create new topics (first-topic) and evaluate the message.

8. Enter following command to create new topics (first-topic) and evaluate the message.

   kafka-topics --zookeeper 127.0.0.1:2181 --topic first-topic --create --partitions 3 --replication-factor 1

9. Wow, Congratulation! Finally you created topic called first-topic with number of partitions are 3 and replication factor is 1 (since it is running on single broker)

10. Enter following command to list out topics you have created
    kafka-topics --zookeeper 127.0.0.1:2181 --list

11. Display details about topics (including number of partitions and replication factor)
    kafka-topics --zookeeper 127.0.0.1:2181 --topic test --describe

12. If required, you can delete topic using following command
    kafka-topics --zookeeper 127.0.0.1:2181 --topic first-topic --delete

Producing Messages using console-producer - Steps

1. By "kafka-console-producer.sh" or "kafka-console-producer.bat" based on your system (or bin/kafka-console-producer.sh or bin\windows\kafka-console-producer.bat if you didn't setup PATH / Environment variables, If you have already set up a path variable, then just type "kafka-console-producer from any directory)

   kafka-console-producer

2. Enter following command to produce messages

   kafka-console-producer --broker-list 127.0.0.1:9092 --topic first-topic
   <Enter following message>
   hello Dr.Bhimappa Desai
   This is first message of apache kafka
   This is second message of apache kafka
   just another message

3. Producing message with properties to get acknowledgement

   kafka-console-producer --broker-list 127.0.0.1:9092 --topic first_topic --producer-property acks=all

   <Enter following message>
   some New message that should acknowledged
   This is just for fun

4. Enter Control + C to stop message at any time.

5. You can produce message to a non existing topic also

   kafka-console-producer --broker-list 127.0.0.1:9092 --topic new_topic

   <Enter following message>
   hello world!

6. Usually new partition has one partition by default, learner can list them topics and also details about topics;

   kafka-topics --zookeeper 127.0.0.1:2181 –list

   kafka-topics --zookeeper 127.0.0.1:2181 --topic new_topic --describe

7. You can edit **num.partitions** property in config/server.properties with with following value;
   # num.partitions=3

8. Again you can produce against a non existing topic and check number of partitions

   kafka-console-producer --broker-list 127.0.0.1:9092 --topic new_topic_2

   <Enter following message>
   hello again!

9. This time our topic has 3 partitions
   kafka-topics --zookeeper 127.0.0.1:2181 --list
   kafka-topics --zookeeper 127.0.0.1:2181 --topic new_topic_2 --describe

10. Overall, please create topics before producing to them!

Consuming Messages using console-consumer (from other terminal) - Steps

1. By "kafka-console-consumer.sh" or "kafka-console-consumer.bat" based on your system. (or bin/kafka-console-consumer.sh or bin\windows\kafka-console-consumer.bat if you didn't setup PATH / Environment variables). If you have already set up a path variable, then just type "kafka-console-producer from any directory)
   kafka-console-consumer

2. Consume current message from the first-topic

   kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic first-topic

Note: here you may not get current message unless you send message instantly from another terminal.

3. From other terminal produce once again message to first-topic

   kafka-console-producer --broker-list 127.0.0.1:9092 --topic first_topic

4. Enter following command to consume message from first-topic from the beginning.

   kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic word-count-input --from-beginning

Consuming Messages using consumer group - Steps

1. By "kafka-console-consumer.sh" or "kafka-console-consumer.bat" based on your system. (or bin/kafka-console-consumer.sh or bin\windows\kafka-console-consumer.bat if you didn't setup PATH / Environment variables). If you have already set up a path variable, then just type "kafka-console-producer from any directory)

2. Start one consumer

kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic first-topic --group my-first-application

3. Start one producer and start producing

kafka-console-producer --broker-list 127.0.0.1:9092 --topic first-topic

4. Start another consumer (from another terminal) part of the same group. See messages being spread

kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic first-topic --group my-first-application --from-beginning

5.Start another consumer(from another terminal so by now you have different consumer with the same consumer application group) part of a different group from beginning

kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic first-topic --group my-second-application --from-beginning

Note: you have 3 different consumer in 23 separate terminal and one producer (total terminals are 4). All 3 consumers are part of same consumer group). Whatever message is being produced it will appear in all 3 consumer terminal (kind of parallel you can process)

Shifting Topic Off-set - Steps

1. By *"kafka-consumer-groups.sh" or "kafka-consumer-groups.bat"* based on your system. (or *bin/kafka-consumer-groups.sh or bin\windows\kafka-consumer-groups.bat* if you didn't setup PATH / Environment variables). If you have already set up a path variable, then just type ***"kafka-consumer-groups"*** from any directory)

2. Enter following command to know the documentation and parameters options of ***kafka-consumer-groups***

3. Enter following command to list out existing consumer groups

kafka-consumer-groups --bootstrap-server localhost:9092 --list

4. Enter following command to describe one specific group

kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group my-second-application

5. Enter following command to describe another group

kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group my-first-application

6. Enter following command to start a consumer

kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic first-topic --group my-first-application --from-beginning

7. Enter following command to describe the group now

kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group my-first-application

Note: Here you will see your are pointing to last off-set.

8. Enter following command to reset the offsets to the beginning of each partition and evaluate the message if you do not get anything then go to step 9.

kafka-consumer-groups --bootstrap-server localhost:9092 --group my-first-application --reset-offsets --to-earliest

9. Enter following by setting **execute** flag and evaluate the message if you do not get anything then go to step 10.

kafka-consumer-groups --bootstrap-server localhost:9092 --group my-first-application --reset-offsets --to-earliest --execute

10. Enter following command since **topic** flag is also needed

kafka-consumer-groups --bootstrap-server localhost:9092 --group my-first-application --reset-offsets --to-earliest --execute --topic first-topic

11. Enter the following command to consume from where the offsets have been reset

kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic first-topic --group my-first-application

12. Enter the following command to describe the group again

kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group my-first-application

13. Enter the following commands to shift off-set (forward and backword from the current off-set.

shift offsets by 2 (forward) as another strategy

kafka-consumer-groups --bootstrap-server localhost:9092 --group my-first-application --reset-offsets --shift-by 2 --execute --topic first-topic

shift offsets by 2 (backward) as another strategy

kafka-consumer-groups --bootstrap-server localhost:9092 --group my-first-application --reset-offsets --shift-by -2 --execute --topic first-topic

consume again

kafka-console-consumer --bootstrap-server 127.0.0.1:9092 --topic first-topic --group my-first-application

Working with Multi-Broker Cluster - Steps

1. In order to work multi node (multi broker) cluster on existing system (desktop/laptop) follow below steps;

2. Copy server.properties into 3 files such as

   # server0.properties and

   # server1.properties and

   # server2.properties

3. Edit each file with following properties:

   # server1.properties

   # broker.id=1

   # listeners=PLAINTEXT://:9093

   # logs.dir=/path/to/data/kafka1

   # server2.properties

   # broker.id=2

   # listeners=PLAINTEXT://:9094

   # logs.dir=/path/to/data/kafka2

4. Then start kafka (3 different terminals)

   kafka-server-start config/server.properties

   kafka-server-start config/server2.properties

   kafka-server-start config/server3.properties

5. you can create a topic with RF of 3!

   kafka-topics --zookeeper 127.0.0.1:2181 --create --topic second-topic --replication-factor 3 --partitions 3

Working with Topic Configuration - Steps

1. By "kafka-topics.sh" or "kafka-topics.bat" based on your system. (or bin/kafka-topics.sh or bin\windows\kafka-topics.bat if you didn't setup PATH / Environment variables). If you have already set up a path variable, then just type "kafka-topics" from any directory).

2. Enter following command to list out topics

   kafka-topics --zookeeper 127.0.0.1:2181 --list

3. Create topic a topic (topic name configure) that can used for configuration purpose.
   kafka-topics --zookeeper 127.0.0.1:2181 --create --topic configured-topic --partitions 3 --replication-factor 1

4. Enter following command to know existing configuration of "configuration" topic.

   kafka-topics --zookeeper 127.0.0.1:2181 --describe --topic configured-topic

5. Check documentation of kafka-configs command

   kafka-configs

6. Enter command to describe configs for the topic

kafka-configs --zookeeper 127.0.0.1:2181 --entity-type topics --entity-name configured-topic --describe

7. Enter command to add configuration to existing topic

   kafka-configs --zookeeper 127.0.0.1:2181 --entity-type topics --entity-name configured-topic --add-config min.insync.replicas=2 --alter

8. Enter command to describe configs using kafka-configs

   kafka-configs --zookeeper 127.0.0.1:2181 --entity-type topics --entity-name configured-topic --describe

9. Describe configs using kafka-topics

   kafka-topics --zookeeper 127.0.0.1:2181 --describe --topic configured-topic

10. Enter Command to Delete a config

    kafka-configs --zookeeper 127.0.0.1:2181 --entity-type topics --entity-name configured-topic --delete-config min.insync.replicas --alter

11. Enter Command to ensure the config has been deleted

    kafka-topics --zookeeper 127.0.0.1:2181 --describe --topic configured-topic

Working with Apache Kafka Streaming (Windows Only) - Steps

(Note: Learners are expected to copy all commands in notepad and keep them in single line and past in terminal to avoid error)

1. Enter Command to Start - zookeeper is at localhost:2181

bin\windows\zookeeper-server-start.bat config\zookeeper.properties

2. Enter command to start (another shell) - kafka is at localhost:9092

bin\windows\kafka-server-start.bat config\server.properties

3. Enter following command to create input topic

bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic streams-plaintext-input

4. Enter following command to create output topic

bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic streams-wordcount-output

5. Enter following command to produce message in (streams-plaintext-input) topic.

bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic streams-plaintext-input

6. Enter following command to verify data written in streams-plaintext-input topic

bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic streams-plaintext-input --from-beginning

7. Enter following command to start a consumer on the output topic
(paste below command in notepad and paste in terminal to avoid error)

bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic streams-wordcount-output --from-beginning –formatter kafka.tools.DefaultMessageFormatter --property print.key=true --property print.value=true --property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer --property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer

8. Enter following command to start the streams application

bin\windows\kafka-run-class.bat org.apache.kafka.streams.examples.wordcount.WordCountDemo

9. Verify the data has been written to the output topic!

About Capgemini

A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2017 global revenues of EUR 12.8 billion.

Learn more about us at www.capgemini.com.