Kafka

Start zookeeper first and then Kafka

To run Zookeeper:

In Terminal goto /Users/parvathichinthalapati/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/bin

and then run below command

./zookeeper-server-start.sh /Users/parvathichinthalapati/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/config/zookeeper.properties

or

Command to start zookeeper

/zookeeper-server-start.sh ~/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/config/zookeeper.properties

To run Kafka server: goto bin folder

./kafka-server-start.sh /Users/parvathichinthalapati/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/config/server.properties

or

In another terminal run Kafka

Command to start Kafka

./kafka-server-start.sh ~/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/config/server.properties

Set PATH:

run below command to open .zshrc file. Its hidden file. To view hidden files run ls -a.

nano .zshrc

add below line and then ctrl + X and save

PATH="$PATH:/Users/parvathichinthalapati/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/bin"

echo $PATH

------------

**Run Kafka using homebrew:**

Install Homebrew using below command:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

Install Kafka using homebrew

brew install Kafka

Start zookeeper using home-brew

/opt/homebrew/bin/zookeeper-server-start /opt/homebrew/etc/zookeeper/zoo.cfg

To start Kafka

/opt/homebrew/bin/kafka-server-start /opt/homebrew/etc/kafka/server.properties

**Start Kafka in Kraft mode without Zookeeper:**

In the Home directory run below command

parvathichinthalapati@Parvathis-M2-MacBook-Pro ~ % kafka-storage random-uuid

3babiyLYQgOhhFXUVaz5BA

This is the cluster id.

format directory:

parvathichinthalapati@Parvathis-M2-MacBook-Pro ~ % ~/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/bin/kafka-storage.sh format -t 3babiyLYQgOhhFXUVaz5BA-c ~/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/config/kraft/server.properties

Run Kafka:

parvathichinthalapati@Parvathis-M2-MacBook-Pro ~ % kafka-server-start ~/Technical\_Stuff/Softwares/kafka\_2.12-3.4.0/config/kraft/server.properties

**Kafka CLI**

**create playground.config file and paste configuration**

**create kafka topics** ->in home folder run brew install kafka

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --create --topic first\_topic

**create topic with partitions:**

kafka-topics.sh --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --create --topic second\_topic --partitions 5

**create topic with replication factor:**

kafka-topics.sh --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --create --topic third\_topic --replication-factor 2

**# list topics**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --list

**# describe topics**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --describe

**# delete topics**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic second\_topic --delete

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

##### LOCALHOST #####

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

kafka-topics.sh

kafka-topics.sh --bootstrap-server localhost:9092 --list

kafka-topics.sh --bootstrap-server localhost:9092 --topic first\_topic --create

kafka-topics.sh --bootstrap-server localhost:9092 --topic second\_topic --create --partitions 3

kafka-topics.sh --bootstrap-server localhost:9092 --topic third\_topic --create --partitions 3 --replication-factor 2

# Create a topic (working)

kafka-topics.sh --bootstrap-server localhost:9092 --topic third\_topic --create --partitions 3 --replication-factor 1

# List topics

kafka-topics.sh --bootstrap-server localhost:9092 --list

# Describe a topic

kafka-topics.sh --bootstrap-server localhost:9092 --topic first\_topic --describe

# Delete a topic

kafka-topics.sh --bootstrap-server localhost:9092 --topic first\_topic --delete

# (only works if delete.topic.enable=true)

**Kafka Produer:**

**create topic**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --create --partitions 1

**Produce Kafka messages:**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic

> Hello World

> just for fun

> fun learning!

**# producing with properties**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --producer-property acks=all

> some message that is acked

> just for fun

> fun learning!

**# producing to a non existing topic**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic new\_topic

> hello world!

It will print error msgs due to non existing topics

**# if topic auto creation is disabled (by default on playground), then the topic will not appear in the list**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --list

**# producing to a non existing topic**

kafka-console-producer --bootstrap-server localhost:9092 --topic new\_topic

> hello world!

[2023-04-25 20:44:03,450] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id 4 : {new\_topic=UNKNOWN\_TOPIC\_OR\_PARTITION} (org.apache.kafka.clients.NetworkClient)

# our new topic only has 1 partition

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

kafka-topics --bootstrap-server localhost:9092 --topic new\_topic --describe

# edit config/server.properties or config/kraft/server.properties

# num.partitions=3

# produce against a non existing topic again

kafka-console-producer --bootstrap-server localhost:9092 --topic new\_topic\_2

hello again!

# this time our topic has 3 partitions

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

kafka-topics --bootstrap-server localhost:9092 --topic new\_topic\_2 --describe

# overall, please create topics with the appropriate number of partitions before producing to them!

# produce with keys

kafka-console-producer --bootstrap-server localhost:9092 --topic first\_topic --property parse.key=true --property key.separator=:

>example key:example value

>name:Stephane

**kafka-console-producer.sh**

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

### CONDUKTOR PLATFORM #####

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

**Create Topic first\_topic with partition1**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --create --partitions 1

**Produce messages in kafka topic**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic

**# producing with properties**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --producer-property acks=all

**# producing to a non existing topic**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic new\_topic

> hello world!

[2023-05-03 13:26:24,834] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id 31 : {new\_topic=UNKNOWN\_TOPIC\_OR\_PARTITION} (org.apache.kafka.clients.NetworkClient)

^C[2023-05-03 13:26:25,054] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id 32 : {new\_topic=UNKNOWN\_TOPIC\_OR\_PARTITION} (org.apache.kafka.clients.NetworkClient)

It gives error because new\_topic does n’t exists.

**# if topic auto creation is disabled (by default on playground), then the topic will not appear in the list**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --list

**# overall, please create topics with the appropriate number of partitions before producing to them!**

**# produce with keys**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --property parse.key=true --property key.separator=:

>example key:example value

>name:Stephane

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

##### LOCALHOST #####

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

**#Create Topics:**

kafka-topics --bootstrap-server localhost:9092 --topic test\_topic --create --partitions 1

**# producing**

kafka-console-producer --bootstrap-server localhost:9092 --topic test\_topic

> Hello World

>My name is Conduktor

>I love Kafka

>^C (<- Ctrl + C is used to exit the producer)

**# producing with properties**

kafka-console-producer --bootstrap-server localhost:9092 --topic test\_topic --producer-property acks=all

> some message that is acked

> just for fun

> fun learning!

**# producing to a non existing topic**

kafka-console-producer --bootstrap-server localhost:9092 --topic new\_topic

> hello world!

**# our new topic only has 1 partition**

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

kafka-topics --bootstrap-server localhost:9092 --topic new\_topic --describe

**# overall, please create topics with the appropriate number of partitions before producing to them!**

**# produce with keys**

kafka-console-producer --bootstrap-server localhost:9092 --topic first\_topic --property parse.key=true --property key.separator=:

>example key:example value

>name:Stephane

**Kafka Consumer:**

**# create a topic with 3 partitions**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic second\_topic --create --partitions 3

**# consuming**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic second\_topic

**# other terminal**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --producer-property partitioner.class=org.apache.kafka.clients.producer.RoundRobinPartitioner --topic second\_topic

**# consuming from beginning**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic second\_topic --from-beginning

**# display key, values and timestamp in consumer**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic second\_topic --formatter kafka.tools.DefaultMessageFormatter --property print.timestamp=true --property print.key=true --property print.value=true --property print.partition=true --from-beginning

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

##### LOCALHOST #####

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

**# create a topic with 3 partitions**

kafka-topics --bootstrap-server localhost:9092 --topic second\_topic --create --partitions 3

**# consuming**

kafka-console-consumer --bootstrap-server localhost:9092 --topic second\_topic

**# other terminal**

kafka-console-producer --bootstrap-server localhost:9092 --producer-property partitioner.class=org.apache.kafka.clients.producer.RoundRobinPartitioner --topic second\_topic

**# consuming from beginning**

kafka-console-consumer --bootstrap-server localhost:9092 --topic second\_topic --from-beginning

**# display key, values and timestamp in consumer**

kafka-console-consumer --bootstrap-server localhost:9092 --topic second\_topic --formatter kafka.tools.DefaultMessageFormatter --property print.timestamp=true --property print.key=true --property print.value=true --property print.partition=true --from-beginning

**Kafka Consumers in Groups**

**# create a topic with 3 partitions**

kafka-topics --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic third\_topic --create --partitions 3

**# start one producer and start producing**

kafka-console-producer --producer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --producer-property partitioner.class=org.apache.kafka.clients.producer.RoundRobinPartitioner --topic third\_topic

**# start one consumer**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic third\_topic --group my-first-application

**# start another consumer part of the same group. See messages being spread**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic third\_topic --group my-first-application

**# start another consumer part of a different group from beginning**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic third\_topic --group my-second-application --from-beginning

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

##### LOCALHOST #####

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

**# create a topic with 3 partitions**

kafka-topics --bootstrap-server localhost:9092 --topic third\_topic --create --partitions 3

**# start one consumer**

kafka-console-consumer --bootstrap-server localhost:9092 --topic third\_topic --group my-first-application

**# start one producer and start producing**

kafka-console-producer --bootstrap-server localhost:9092 --producer-property partitioner.class=org.apache.kafka.clients.producer.RoundRobinPartitioner --topic third\_topic

**# start another consumer part of the same group. See messages being spread**

kafka-console-consumer --bootstrap-server localhost:9092 --topic third\_topic --group my-first-application

**# start another consumer part of a different group from beginning**

kafka-console-consumer --bootstrap-server localhost:9092 --topic third\_topic --group my-second-application --from-beginning

**Kafka Consumer Groups CLI:**

**# list consumer groups**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --list

**# describe one specific group**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group my-second-application

**# describe another group**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group my-first-application

**# start a consumer**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --group my-first-application

**# describe the group now**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group my-first-application

**# describe a console consumer group (change the end number)**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group console-consumer-10592

**# start a console consumer**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic first\_topic --group my-first-application

**# describe the group again**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group my-first-application

**############################**

**##### LOCALHOST #####**

**############################**

**# list consumer groups**

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

**# describe one specific group**

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

**# describe another group**

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

**# start a consumer**

kafka-console-consumer --bootstrap-server localhost:9092 --topic first\_topic --group my-first-application

**# describe the group now**

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

**# describe a console consumer group (change the end number)**

kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group console-consumer-10592

**# start a console consumer**

kafka-console-consumer --bootstrap-server localhost:9092 --topic first\_topic --group my-first-application

**# describe the group again**

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

**Resetting offsets:**

**# describe the consumer group**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group my-first-application

**# Dry Run: reset the offsets to the beginning of each partition**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --group my-first-application --reset-offsets --to-earliest --topic third\_topic --dry-run

**# execute flag is needed**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --group my-first-application --reset-offsets --to-earliest --topic third\_topic --execute

**# describe the consumer group again**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group my-first-application

**# consume from where the offsets have been reset**

kafka-console-consumer --consumer.config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --topic third\_topic --group my-first-application

**# describe the group again**

kafka-consumer-groups --command-config playground.config --bootstrap-server cluster.playground.cdkt.io:9092 --describe --group my-first-application

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

##### LOCALHOST #####

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

**# describe the consumer group**

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

**# Dry Run: reset the offsets to the beginning of each partition**

kafka-consumer-groups --bootstrap-server localhost:9092 --group my-first-application --reset-offsets --to-earliest --topic third\_topic --dry-run

**# execute flag is needed**

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

**# describe the consumer group again**

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

**# consume from where the offsets have been reset**

kafka-console-consumer --bootstrap-server localhost:9092 --topic third\_topic --group my-first-application

**# describe the group again**

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