

Kafka commands

On the Cluster navigate to the /home/ubuntu/kafka directory then use the kafka sh files in the bin directory to execute the commands below. You might need to be root in order to run some commands.

Most of commands need "bootstrap-server" property that can take comma separated list of broker:port
e.g. kafka3:9092,kafka2:9092,kafka1:9092

Topics

Create:

```
./bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic <name-of-topic> --partitions 10 --replication-factor 3 --create
```

```
./bin/kafka-topics.sh --bootstrap-server kafka3:9095,kafka2:9094,kafka1:9092 --create --if-not-exists --topic lndcdcdsprpsl_flightrangel00partitions --partitions 100 --replication-factor 3 --config min.insync.replicas=2 --config cleanup.policy=compact
```

Describe:

```
./bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic <name-of-topic> --describe
```

Check offsets:

```
./bin/kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list localhost:9092 --topic <name-of-topic>
```

Latest: `./bin/kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list localhost:9092 --topic <name-of-topic> --time -1`

Earliest: `./bin/kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list localhost:9092 --topic <name-of-topic> --time -2`

List:

```
./bin/kafka-topics.sh --bootstrap-server localhost:9092 --list
```

Delete:

```
./bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic <name-of-topic> --delete
```

```
./bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic <name-of-topic> --delete --force
```

Script to delete matching topics

```
for my_topic in $(./bin/kafka-topics.sh --bootstrap-server localhost:9092 --list | grep '^lndcdcds' | sort);  
do  
    ./bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic $my_topic --delete --force;  
    echo "Deleted topic: " $my_topic;  
done;
```

Consumer Groups

List:

To check the latest offsets and lags by partition for the given consumer group. Note: for generic kafka-consumer, the "client.id" acts as consumer group. For kafka-streams or ksqldb application, the consumer group is dynamically created from "application.id"

```
./bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --group <consumer-group-id> --describe
```

To get all consumers connected :?

```
./bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --all-groups --list
```

Move offsets: ? ⚠

Note: *Exercise caution* while moving offsets manually may cause other consequences in behaviour of the existing attached consumer group(s). Dry run to confirm intended result, before executing the command.

An easier option could be to delete the topic and recreate it using the same properties as earlier.💡

```
dry run: ./bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --topic <topic-name> --group
<consumer-group-id> --reset-offsets? --to-earliest --dry-run
```

```
execute: ./bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --topic <topic-name> --group
<consumer-group-id> --reset-offsets? --to-earliest --execute
```

Empty topic or Delete all messages:?!⚠️

1. Note the properties of the existing topic using --describe option.
2. run: `./bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type topics --entity-name <topic-name> --alter --add-config retention.ms=100`
3. Wait for a minute or so and ensure the messages are deleted from topic
4. Then, issue above command now with original retention.ms configuration to restore

Delete consumer-group:

1. Check if the consumer-group is empty: `./bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --group? <consumer-group-id> --describe --state`
2. `./bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --group?<consumer-group-id> --delete?`

Utility consumers for investigation

KSQLDB:

On ksqldb prompt, use `print <topic-name>;` or `print <topic-name> from beginning;`

Console consumer:

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

KafkaCat:

```
brew install kafkacat
```

```
kcat -C -b localhost:9092 -t lndcdcadsvntry_exclsvtytyp -r http://localhost:8081 -s key=s -s value=avro -o
beginning -c 3 -e
```